

**INDIVIDUAL PILL MONITORING SYSTEM FOR IMPROVED MEDICATION  
ADHERENCE ACCURACY**

by

Brian Babes Venus

BS, University of Pittsburgh, 2006

Submitted to the Graduate Faculty of

The Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2008

UNIVERSITY OF PITTSBURGH  
SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Brian Babes Venus

It was defended on

November 25, 2008

and approved by

James T. Cain, Professor Emeritus, Department of Electrical and Computer Engineering

J. Robert Boston, Professor, Department of Electrical and Computer Engineering, Department

of Bioengineering, Department of Communications Science and Disorders

Ronald G. Hoelzeman, Associate Professor, Department of Electrical and Computer

Thesis Advisor: Marlin H. Mickle, Nickolas A. DeCecco Professor, Department

of Electrical and Computer Engineering

Copyright © by Brian Babes Venus

2008

# **INDIVIDUAL PILL MONITORING SYSTEM FOR IMPROVED MEDICATION ADHERENCE ACCURACY**

Brian Babes Venus, MS

University of Pittsburgh, 2008

Adherence to medication prescription schedules is an important step in the prevention and treatment of diseases in modern medicine. For the most part, adherence is the responsibility of the individual or their caregivers requiring them to remember to take the proper doses every day. Missing doses of critical medications can affect the overall health of the individual and drastically set back the progress of the treatment wasting both the time and resources of the medical community. Electronic solutions exist to aid in this process by reminding, measuring, and alerting individuals of their medication schedules, and help to keep detailed logs to verify overall patient adherence. While each of these solutions has improved overall medication adherence, there are still reports of difficulties with their use and accuracy of the results.

A new battery powered portable smart pillbox concept has been designed to address the issues with current solutions, focusing particularly on ease of use and accuracy. A proof of concept prototype system has been developed using a commercial of the shelf (COTS) pillbox form factor demonstrating both the operation and feasibility of the design. This system utilizes pills distributed in blister packs, which is a common and familiar packaging technology with many benefits. Multiple pill detection strategies were considered, and an innovative optical solution was developed that could determine when individual pills were removed from the packaging increasing adherence accuracy. The pillbox stores a complete time log of pill removal events, and transmits the record wirelessly to receiving stations. Wireless Radio Frequency (RF)

communication is implemented utilizing protocols similar to active Radio Frequency Identification (RFID) standards. This demonstrates the ability of the pillbox to utilize possible future RFID receiver infrastructures in hospitals, pharmacies, and homes expanding the functionality of such RFID systems. A prototype receiving station was developed as a testing platform for the pillbox verifying the systems overall operation and providing a baseline Human Machine Interface (HMI). The final system provides successful pill monitoring with a device that appears both physically and functionally to be a normal pillbox, improving the chances of its acceptance and use by patients.

## TABLE OF CONTENTS

<b>PREFACE.....</b>	<b>XIV</b>
<b>1.0 INTRODUCTION.....</b>	<b>1</b>
<b>1.1 CURRENT MEDICATION ADHERENCE SOLUTIONS.....</b>	<b>2</b>
<b>1.1.1 Medication Event Monitoring System (MEMS).....</b>	<b>3</b>
<b>1.1.2 Med-eMonitor .....</b>	<b>3</b>
<b>1.1.3 MD.2.....</b>	<b>4</b>
<b>1.1.4 MedTracker .....</b>	<b>4</b>
<b>1.1.5 uBox .....</b>	<b>4</b>
<b>1.2 LIMITATIONS OF CURRENT SOLUTIONS .....</b>	<b>5</b>
<b>2.0 PROBLEM STATEMENT .....</b>	<b>7</b>
<b>2.1 REQUIREMENTS AND SPECIFICATIONS.....</b>	<b>7</b>
<b>2.2 PACKAGING DESIGN REFERENCE .....</b>	<b>9</b>
<b>2.3 RECORD KEEPING AND COMMUNICATION .....</b>	<b>10</b>
<b>2.4 POWER CONSIDERATIONS.....</b>	<b>11</b>
<b>2.5 DEVICE OPERATION.....</b>	<b>12</b>
<b>3.0 DESIGN .....</b>	<b>13</b>
<b>3.1 PILLBOX PHYSICAL CONSTRAINTS.....</b>	<b>14</b>
<b>3.1.1 Pillbox Reference .....</b>	<b>14</b>

3.1.2	Pillbox Modifications.....	16
3.2	PILL DETECTION TECHNIQUES .....	20
3.2.1	Push Through Tracks.....	20
3.2.2	Infrared (IR) Interrupters .....	22
3.2.3	IR Sandwiched Array.....	23
3.3	PILL DETECTION DESIGN.....	25
3.3.1	Operation.....	25
3.3.2	IR Emitters.....	26
3.3.3	IR Sensors.....	29
3.3.4	Component Testing.....	30
3.3.5	Event Trigger .....	32
3.4	MICROCONTROLLER.....	33
3.4.1	Selection.....	34
3.4.2	Limitations .....	35
3.4.3	Initialization and Configuration.....	37
3.4.4	Overview.....	38
3.4.5	Real Time Clock.....	40
3.4.6	Pill Detection .....	42
3.4.7	Clock Regulation.....	43
3.4.8	Battery Monitoring.....	46
3.4.9	RF Communication .....	47
3.4.9.1	Standards .....	48
3.4.9.2	Packet Format .....	49

3.4.9.3	Communication Program Flow .....	50
3.5	RF INTEGRATED CIRCUIT .....	53
3.5.1	Configuration .....	54
3.5.2	Antenna Matching Network .....	55
3.6	POWER CONSIDERATIONS .....	58
3.7	RECEIVING STATION .....	59
3.7.1	Real Time Spectrum Analyzer .....	59
3.7.2	LabVIEW .....	62
3.7.2.1	Logic Level Data Stream Recovery .....	62
3.7.2.2	Manchester Decoding .....	64
3.7.2.3	Data Presentation .....	67
4.0	PROTOTYPE ASSEMBLY .....	69
4.1	BOARD LAYOUT .....	70
4.2	ASSEMBLY .....	73
5.0	FUNCTIONAL TESTING .....	76
5.1	EMPTY PILLBOX .....	76
5.2	REFILL CONDITION .....	77
5.3	FULL PILLBOX .....	79
5.4	ONE PILL REMOVED .....	80
5.5	TWO PILLS REMOVED .....	81
5.6	VOLTAGE MEASUREMENT TEST .....	82
5.7	FUNCTIONAL CONCERNS .....	83
6.0	CONCLUSIONS .....	84



<b>7.0</b>	<b>FUTURE WORK .....</b>	<b>85</b>
	<b>APPENDIX A .....</b>	<b>87</b>
	<b>APPENDIX B .....</b>	<b>93</b>
	<b>APPENDIX C .....</b>	<b>96</b>
	<b>BIBLIOGRAPHY .....</b>	<b>99</b>

## LIST OF TABLES

Table 1. Priority of Interrupts Used for Pillbox.....	35
Table 2. Solutions to the Timer Configuration Equation.....	41

## LIST OF FIGURES

Figure 1. Birth Control, An Example of Monthly Blister Pack Prescriptions .....	9
Figure 2. Blister Pack Design Reference .....	10
Figure 3. Main Functional Components .....	13
Figure 4. Solidworks Model of Original Pillbox .....	15
Figure 5. Modifications to the Inside of Pillbox Top.....	16
Figure 6. Modifications to Inside of Pillbox Bottom.....	17
Figure 7. Modifications to Blister Pack Holder Center Piece.....	18
Figure 8. Assembly of Blister Pack Holder .....	18
Figure 9. Modified Case Assembly with PCBs .....	20
Figure 10. Push Through Tracks Method Emulating Matrix Keypad, Bottom of Blister Pack ...	21
Figure 11. Push Through Tracks Method with Individual Traces for Each Pill, Bottom of Blister Pack.....	22
Figure 12. IR Interrupter Method, Top of Blister Pack .....	23
Figure 13. IR Matrix Method, Cross Sectional View .....	24
Figure 14. Pill Detection Operational Example.....	25
Figure 15. Pill Detection Functional Hardware Connections .....	26
Figure 16. IR Emitter Current Limiting.....	27
Figure 17. Pill Detection Beam Angle Error Example .....	28
Figure 18. Common Emitter Amplifier Configuration .....	29
Figure 19. Photo Transistor Superimposed Network Connection .....	30

Figure 20. Switch De-bounce Circuit .....	33
Figure 21 .Initialization Sequence .....	37
Figure 22. Software Overview .....	39
Figure 23. Real Time Clock Functionality .....	41
Figure 24. Pill Detection Functional Diagram.....	43
Figure 25. Clock Regulation Functional Diagram.....	44
Figure 26. DCO Control Interrupt .....	46
Figure 27. Battery Monitoring Functionality.....	47
Figure 28. ISO 18000 part 7 Timing Diagram for Tag Transmission .....	48
Figure 29. Manchester Encoding Standard.....	49
Figure 30. Transmission Packet Structure .....	50
Figure 31. RF Communication Functional Diagram .....	51
Figure 32. RF Interrupt Functional Details.....	53
Figure 33. Reference RF Output Network for Transmitting.....	56
Figure 34. Designed RF Output Network for Transmitting.....	57
Figure 35. RTSA FM Demodulation .....	60
Figure 36. Receiver Functional Overview .....	62
Figure 37. LabVIEW Interface to RTSA, Top Waveform Raw FM Demodulation, Bottom Bit Stream .....	64
Figure 38. Example Waveform, First Bit Transmitted is a 0.....	65
Figure 39. Example Waveform, First Bit Transmitted is a 1 .....	65
Figure 40. Manchester Decoding Algorithm Using Pulse Widths .....	66
Figure 41. Entire LabVIEW Interface for Receiver.....	68
Figure 42. Disassembled Prototype Pillbox.....	69
Figure 43. Placement of Board Interconnect Header Pins .....	71
Figure 44. MSP430 Programming Connection.....	72

Figure 45. Top Half of Pillbox with Populated PCB .....	74
Figure 46. Bottom Half of Pillbox with Populated PCB .....	75
Figure 47. Function Testing, Empty Pillbox.....	77
Figure 48. Function Testing, Refill First Transmission.....	78
Figure 49. Functional Testing, Full Pillbox .....	79
Figure 50. Functional Testing, 1st Pill Removed .....	80
Figure 51. Functional Testing, 2 <sup>nd</sup> Pill Removed .....	81
Figure 52. Functional Testing, Battery Monitor Validation .....	82

## **PREFACE**

I would like to thank Dr. Mickle and Dr. Cain for their guidance and support through my graduate career. I would not have been able to learn, apply, and grow as much as I have without their support.

## **1.0 INTRODUCTION**

Medication adherence is an important part of the healthcare industry today ensuring the effective treatment of serious and chronic illnesses. Healthcare providers need a reliable way to verify the treatment plans prescribed to patients are being followed. Today, adherence is typically the responsibility of the patient and self reporting is the most common and sometimes only feasible method. The system of self reporting and reliance has many problems since it depends on the reliability and memory of patients. We are all human and tend to forget things from time to time, which can lead to accidental doses being missed or doubled. This problem can escalate with age, which is particularly troublesome since many elderly citizens require daily treatment regimens.

Modern treatments for many serious chronic diseases rely on adherence to be effective where missing a scheduled dose can set drastically set back a treatment plan. The treatment of viruses such as HIV and AIDs greatly depends on the adherence of patients to their treatment regimen where variations in the medication schedule can have disastrous consequences allowing an increase in the rate of progression of the disease. Any technologies which can aid patients in the process of remembering doses and recording adherence for medical records can be of great importance not only in helping to extend and improve the lives of patients, but allowing treatment plans to be improved based on the accurate logs of medication use. By improving adherence, medical resources can be used more efficiently by eliminating the wasted resources resulting from inconsistent treatment plans [2].

Healthcare facilities and their caregivers are responsible for their residents and their treatment regimens. Current adherence systems rely on both the caregivers keeping detailed records and an auditing system to store and verify all of these records. This system is inefficient and time consuming requiring extensive bookkeeping. An automatic medication tracking system would eliminate many of the difficulties with the current system allowing for automatic records to be generated, stored, and transmitted. This is a huge benefit allowing the caregivers to focus more on the residents than the required paperwork.

One of the most critical applications of medication adherence is during clinical research, which is trying to determine the effectiveness and side effects of potential new treatments. These studies must be as controlled and consistent as possible to learn of all the potential side effects of a drug before releasing it to the general public. The more accurate the data is from such a trial, the better the analysis of the overall effectiveness of the treatment in relation to the potential side effects. This is critical as it provides the basic information needed to determine the safety of the new treatment.

## **1.1 CURRENT MEDICATION ADHERENCE SOLUTIONS**

Both commercial and academic solutions currently exist to aid in medication conformance. Each has its benefits and is specialized for certain situations.



### **1.1.1 Medication Event Monitoring System (MEMS)**

MEMS is the most well known commercial solution for medication adherence as it was originally released in the late 80's and has received some slight improvements in recent years. The device is essentially a smart pill bottle cap which detects each time the cap is taken off. The device timestamps each time the bottle is opened and assumes that only one pill is removed each time it is opened. Recent improvements to the device include the integration of an Liquid Crystal Display (LCD) to alert a patient when they should take their next dose [8]. Many studies have been done utilizing the MEMS and analyzing its benefits and disadvantages [3]. The MEMS is not the most compact solution as an entire pill bottle is attached to the device.

### **1.1.2 Med-eMonitor**

The Med-eMonitor is a new product which is just beginning to be released. The device is designed to be used at home and is a general health station which tracks not only your medication usage, but asks you general health questions each day. It sounds reminders of medication schedule time and dispenses the proper dose. The device communicates wirelessly to the internet and provides a safety mechanism to alert both healthcare providers and family members if medication doses are being missed [4]. This device is not designed to be portable, but a part of your home.

### **1.1.3 MD.2**

The MD.2 is similar to the Med-eMonitor as it is meant to be a part of your home. It is about the size of a coffee maker and dispenses pill cups on a schedule with a voice alarm to remind you of the dosing schedule. A monitoring service is also offered with the product to track adherence [5].

### **1.1.4 MedTracker**

The MedTracker is a university project which developed a prototype smart pillbox utilizing a common 7-day pill planner. The device was outfitted with a microcontroller, Bluetooth, and a sensor to detect when one of the compartments of the pillbox was opened. The device timestamps and stores a record of the opening events and transmits the information every two hours using a Bluetooth link to a host computer. The device is not very compact and requires refilling every 7 days, but was a successful design as shown in field tests [7].

### **1.1.5 uBox**

The uBox is another university project being developed at MIT. The uBox is a small compact device that contains 14 doses of medication that is dispensed at scheduled dosing times. The box flashes and buzzes notifying the patient when it is time to take a dose. The design only allows the box to be opened once per dosing time to avoid overdosing and stores the timestamp log of each opening event. The device was specifically designed to help treat tuberculosis in underdeveloped countries improving the effectiveness of treatments [6].

## 1.2 LIMITATIONS OF CURRENT SOLUTIONS

Most of the current solutions are not very compact, which can make their use difficult when not around the house. This is especially true for the larger and more functionally complicated Med-eMonitor and MD.2. A solution that is not portable is much more likely to cause missed doses during busy days away from the home when compared to something that can be easily kept in a pocket. This was one of the functional complaints patients have made with the MEMS solution [3]. In order to be used and reliable, the device must be convenient for the patient to use in any situation.

Because of the structured nature of treatment plans and their schedules, solutions such as 7 day pillboxes are extremely useful for patients to keep track of their own medications. This was one of the reasons the MEMS solution had problems being adopted since it forced patients to stop using their own organizational methods for treatment plans [2]. Disrupting the patterns and routines of patients in a treatment plan is risky and many health care providers would not expose patients to such a situation.

All of the portable devices assume that once a container is opened, the proper dose has been taken out. This is once again a major disadvantage found with the MEMs device as the entire pill bottle is available when it is opened and allows for decanting to occur (removing multiple doses at one time) with no way to track such abuses[3]. The MedTracker and uBox both minimize this issue by monitoring which compartment is opened and limiting the number of pills available in those compartments.

A final disadvantage that all methods experience is the refilling process of the actual device. The most standard package used by any of the methods is the MEMS solution. Each of the other solutions requires specialized containers to be loaded with the proper medication and

the proper doses. This added complexity adds time and cost to these solutions making them less appealing than the simple MEMS. This is likely the reason why the MEMS solution is still a viable commercial option even with its other disadvantages.

## **2.0 PROBLEM STATEMENT**

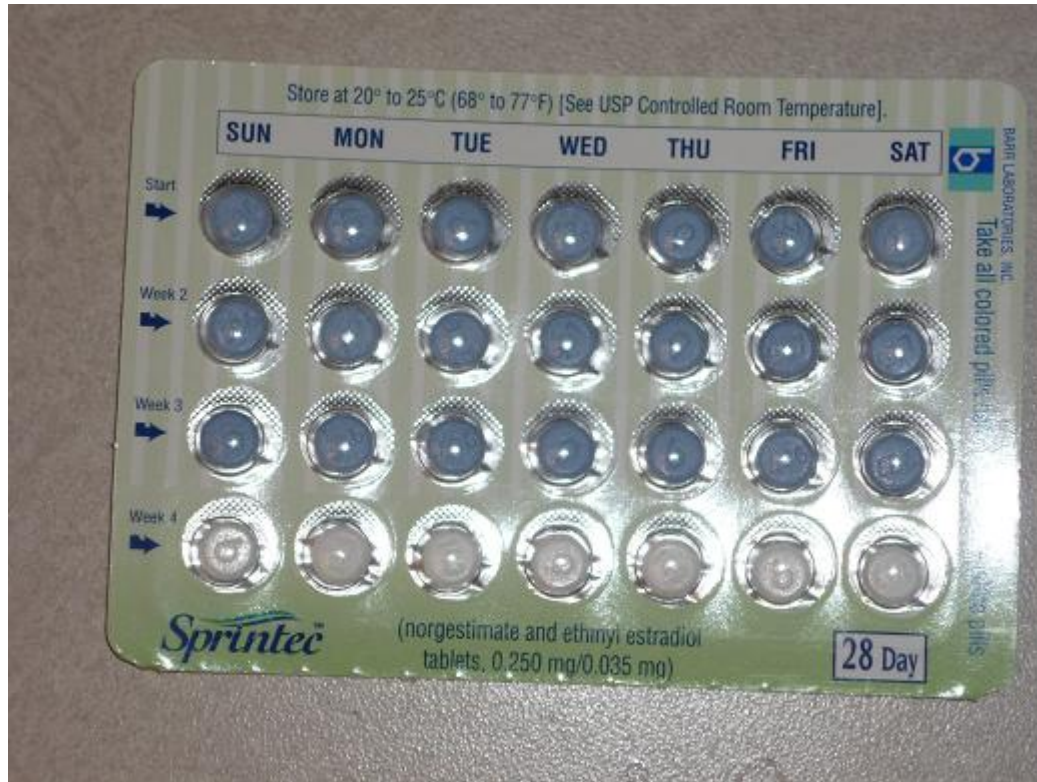
In order to address the shortcomings of all the previously discussed devices, a new portable smart pillbox must be designed and developed to track individual pill removal. This device will utilize common pill distribution technology as a basis to improve adoption of the pillbox and streamline the system for refilling and using the device. The overall use must be simplified so that all patients can easily operate it and remove medication without complications. Simple scheduling routines similar to the 7 day pillbox method should be utilized to compliment current patterns used by patients. The communication of the system will be wireless to minimize complicated interfaces with the device itself utilizing technology which may be common in both pharmacies and homes in the future. The overall goal is to develop a compact and simple system that allows for medication adherence tracking with minimal additional effort from the patient or the pharmacy/health care industry.

## **2.1 REQUIREMENTS AND SPECIFICATIONS**

Besides a pill bottle, the most common pill distribution technology utilized today is blister packs. These packs are a matrix arrangement of pills encased in plastic with formed compartments or blisters to hold each pill. The pills are held into place and sealed by either a thin foil or paper film. This packaging has many benefits including easier drug handling, better shelf life, and

tamper resistance along with reduced packing and shipping costs [1]. The pills are removed by simply applying force to a blister containing the pill and pressing it through the back film. There are variations to this structure to make the technology more child resistant and therefore difficult to open, but optimal ease of use is a primary design goal. Because of this, a simple press through package model is utilized.

This technology is already utilized in monthly prescription applications such as birth control as can be seen in Figure 1 below. They are typically arranged in a matrix of 4 rows and 7 columns to provide a structure matching the 7 days a week and approximately 4 weeks per month, which makes it easier to follow prescription regimens. The blister packs also provide an easy means for labels to be added to the pill compartments to help with time reminders or days of the week as the information can be printed on the film, which is easily seen through the translucent plastic pill encasing. Due to these benefits, this pill packaging technology was chosen to be the basis for this medication tracking system as it inherently satisfies a portion of the problem statement providing a system to compliment patient treatment scheduling.



**Figure 1.** Birth Control, An Example of Monthly Blister Pack Prescriptions

## 2.2 PACKAGING DESIGN REFERENCE

There are no standards dictating the dimensions or shape of blister pack distributions of medication with each manufacturer utilizing their own designs. As such, a simple over the counter blister pack package was utilized for the design and testing of the medication tracker. Figure 2 shows the pills and packaging utilized for this project. These Giant Eagle Brand Nasal Decongestants are relatively inexpensive and easily obtainable at any of the numerous Giant Eagle markets in the Pittsburgh area making them a convenient design reference. Because the blister packs come in 3x3 matrix form, the design will only consider 9 pills, which should provide an adequate proof of concept for this medication tracking method. The hope would be to

expand this method in the future to fit the typical 28 pill monthly prescriptions as discussed earlier. This lack of standards also illustrates an obstacle in the adoption of any method using blister packs as each pillbox would need to be customized to a given manufacturers specifications. This difficulty can be minimized by making such modifications as simple as possible.



**Figure 2.** Blister Pack Design Reference

## **2.3 RECORD KEEPING AND COMMUNICATION**

The device should keep a time record of when each pill is removed providing an automatic log of dose consumption aiding in the aforementioned auditing issue. This log should not be able to be modified unless the device is being refilled by the pharmacy. Each time a pill is removed, the device should attempt to transmit the current log to a receiving station. If a receiving station is not nearby, the device should continue to operate normally and can update the receiving station



log during a later transmission. This functionality is similar to a portable email client synchronizing with the server when in range of an access point. The order in which the pills are removed should not affect the operation of the system and the receiving station should be capable of providing the status of any pill within the case. The transmitted information should also include an ID tag to identify the pillbox and a battery status measurement providing a means for early warning of battery failure.

Low power Radio Frequency (RF) communication should be utilized with RFID standards followed as closely as possible. This is to prepare the device to be compatible with the developing RFID capabilities of both retailer and pharmaceutical companies. It is envisioned that future pharmacies will have standard RFID equipment installed for inventory and quality control of medication, providing a suitable communication method for an adherence system.

## **2.4 POWER CONSIDERATIONS**

The battery life of the device should be able to support use for up to a month. This should be tested by successfully transmitting a message 30 times to simulate the monthly use of the pillbox. This should be an adequate measure of battery life as transmitting is the highest power demand task. As such, the time the device spends transmitting should be minimized in order to optimize battery life.

The device is only required to transmit and not to receive and should be designed to function as such. This is once again to minimize power use as the smallest feasible battery solution is desired in order to provide a compact solution. The device should minimize current

use at all times requiring power efficient electronics and a microcontroller with power management capabilities.

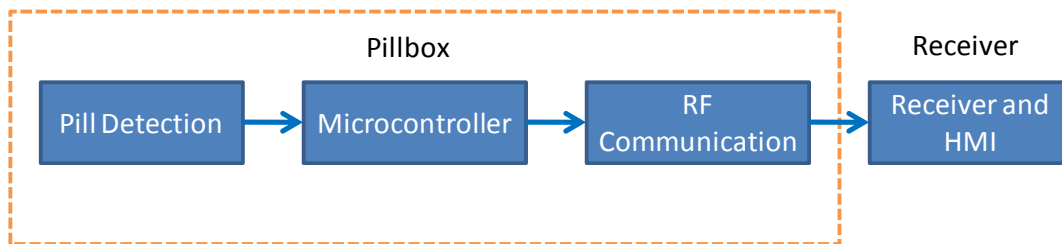
## **2.5 DEVICE OPERATION**

The device should be compact and resemble common pillbox dimensions to allow maximum portability and ease of use. This compact requirement forces the careful selection of components and their arrangement, hence, increasing the design complexity. The complex functionality of the device should be transparent to the patient allowing them to only need to remove the needed medication.

The device should require minimal expertise to operate even by the pharmacy staff. The pharmacy tasks should only include simple battery replacements and refilling the pillbox with the appropriate blister pack. These operations should be quick and straight forward providing automatic methods for updating both the medication records for that pillbox and the pillbox itself. A user friendly and intuitive Human Machine Interface (HMI) for these operations should be available to the pharmacies.

### 3.0 DESIGN

The medication tracking concept requires multiple functional elements to be designed in order to provide a fully operational system. Figure 3 below provides an overview of these main functional elements, how they are connected, and where they are located in the final system.



**Figure 3.** Main Functional Components

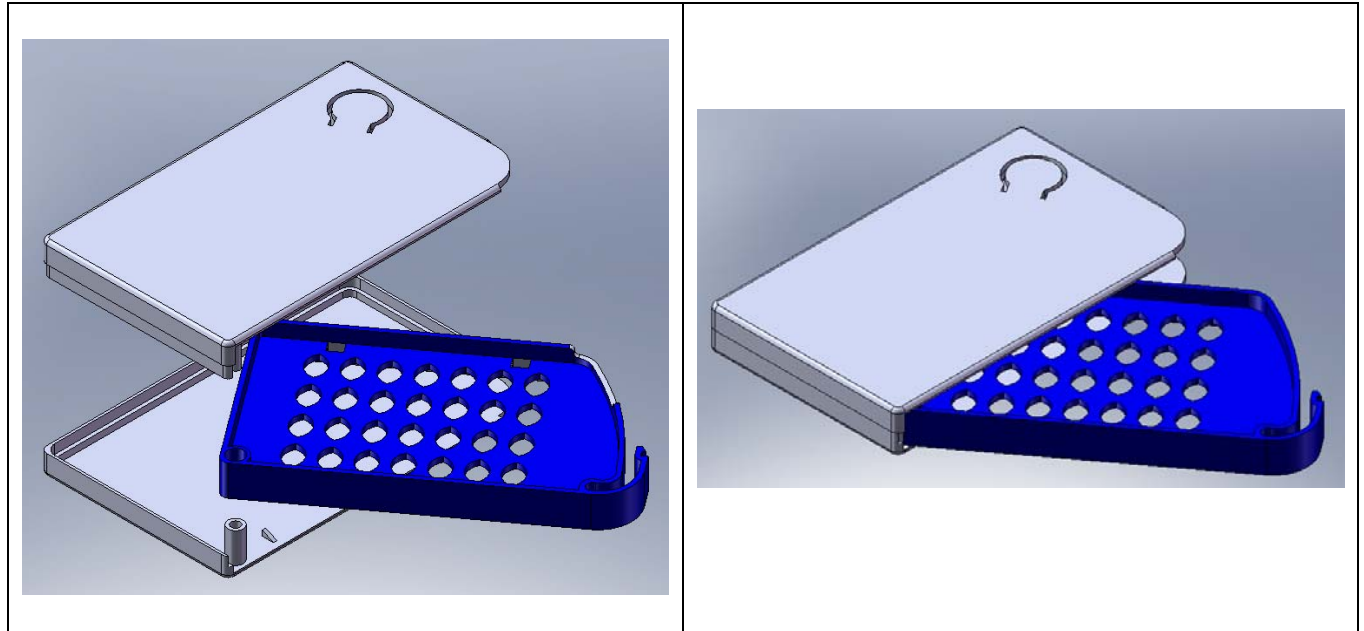
The details of each of these functional elements are discussed in great detail in the following subsections including information on the theory of their operation and relevant design constraints. The pillbox encompasses the majority of the functional elements and is the focus of many of these subsections as it required detailed hardware and software design. The increased number of constraints for the pillbox (compact and battery powered) also added to the increased design complexity.

### **3.1 PILLBOX PHYSICAL CONSTRAINTS**

The size of the pillbox is a critical design constraint as the device must be portable and convenient to use anywhere. An existing blister pack pillbox provided a form factor reference, which was utilized to provide dimensional information. This is critical as it provides information about the available space and optimal locations to place electronic components within the pillbox's physical constraints.

#### **3.1.1 Pillbox Reference**

There are many manufacturers who can design and manufacture pillboxes for blister pack medications. One of these concept designs was obtained by the University and served as the basis for the pillbox case. The case was a commercial off the shelf dispensing box manufactured by West Pharmaceutical. It is constructed completely out of plastic, which is ideal for an RF transmitter as it will not attenuate the generated signal. The first step in designing around this case constraint was to accurately model the case in Solidworks, a 3-D Computer Aided Design (CAD) package, to provide needed measurements and clearances. The resulting Solidworks model is shown in Figure 4 below. This model provided advanced knowledge of the physical relationships of the various parts of the pill case before any modifications were made or Printed Circuit Boards (PCB's) were designed or ordered. This was a powerful advantage as the model was used as the basis for all other design steps.



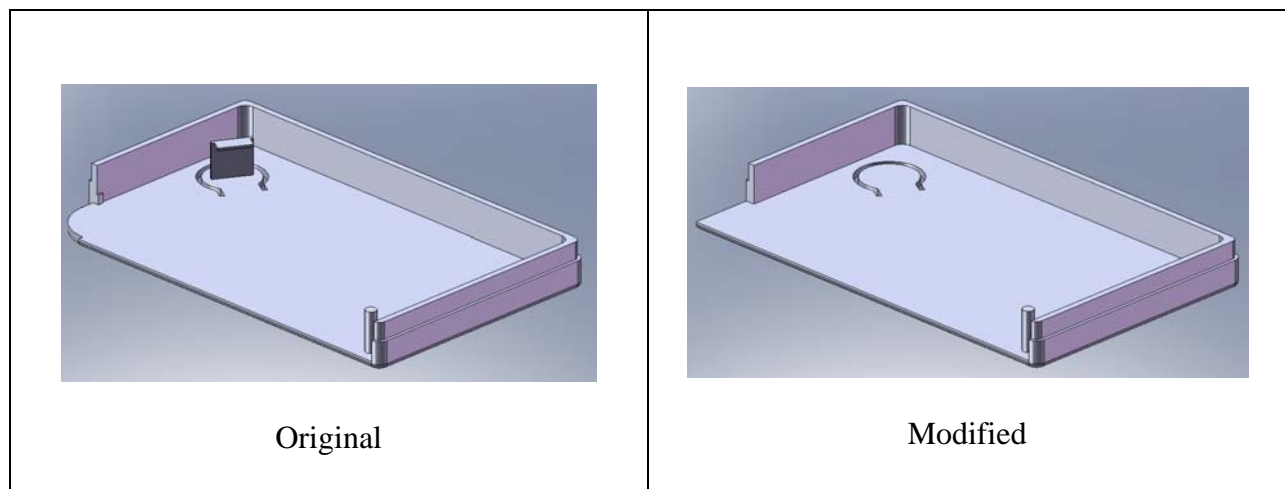
**Figure 4.** Solidworks Model of Original Pillbox

The case is comprised of three sections: a bottom, a top, and a center blister pack mount. The center section slides in and out of the top and bottom of the case by pivoting around a boss between the top and the bottom as shown in the figure. The center piece will lock into a closed position and could only be released by pushing the release button on the top, and pressing in the top corner of the blister pack mount. This two switch functionality was likely added to provide child safety functionality.

The blister pack holder included clips on the top to hold a blister pack in place and holes in which the pills could fall after being pressed through the film backing. The blister pack holder was designed for a 28 pill pack, but as discussed earlier there are no standards for the layout of blister packs or their dimensions which makes finding a matching blister pack near to impossible. In order to utilize this case for the prototype, modifications were required to match the design reference blister pack dimensions and to allow both PCB's and components to fit within the case. These modifications were relatively minor but were first done in Solidworks to provide the needed dimensions streamlining the machining process.

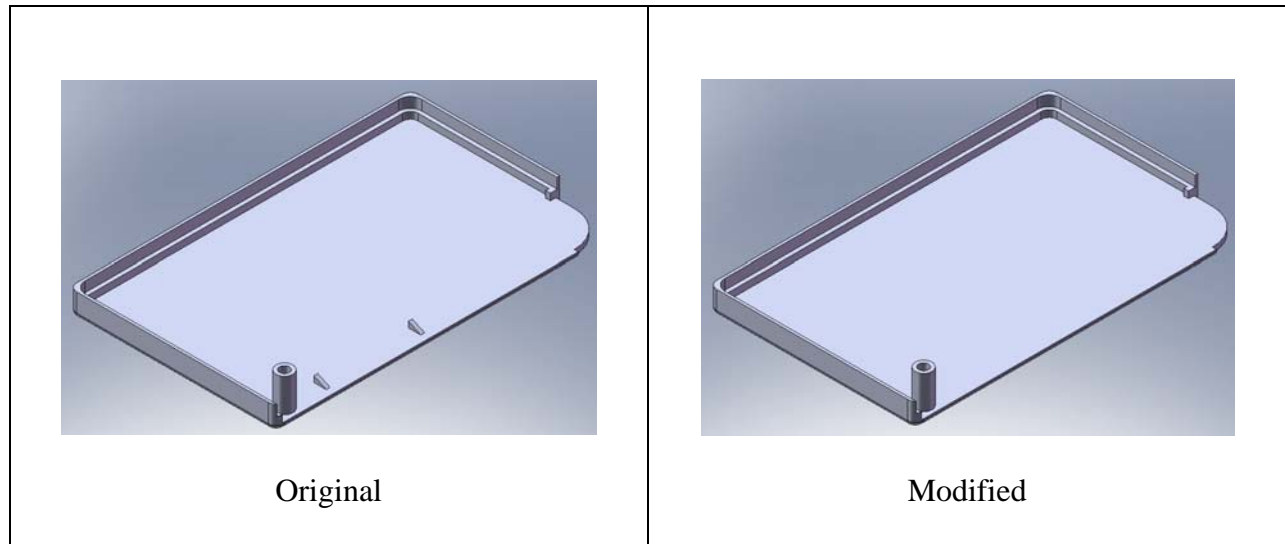
### 3.1.2 Pillbox Modifications

The first modification to the pillbox was to eliminate potential obstacles for PCB's being mounted to the inside of the top and bottom pieces of the case. This included the push button locking mechanism that holds the blister pack holder closed. Because there are two locking mechanisms, this change does not seriously compromise the original operation of the pillbox. The removal of the locking mechanisms is shown in Figure 5.



**Figure 5.** Modifications to the Inside of Pillbox Top

All other obstructions were relatively minor bumps and shapes on the inside of the case, which were easily ground and sanded to a level surface as shown in Figure 6. This leveling process was required in order to allow a PCB to be mounted flush to the case. If the case was not leveled, an unevenly mounted PCB would waste space inside the pillbox and reduce the clearance between the board and blister pack holder.

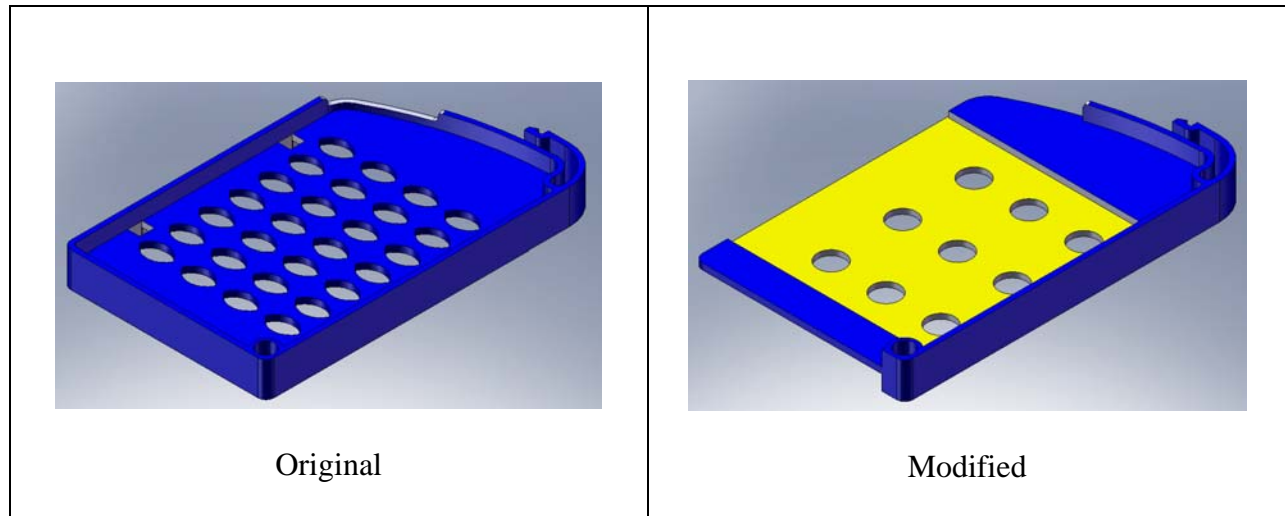


**Figure 6.** Modifications to Inside of Pillbox Bottom

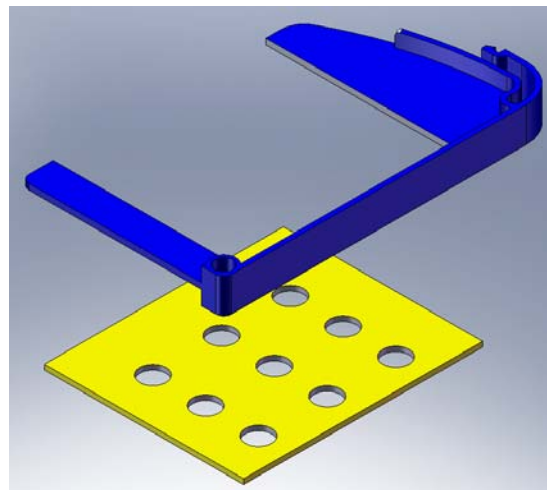
The blister pack holder itself required the most modification. The holder was surrounded by a stiffening rim which would slide across the inside of both the top and bottom of the case. This was an issue as any added obstructions to the inside of the case such as circuit boards or components would prevent the blister pack holder from sliding open or closed. In order to alleviate this issue, the rims of the three internal sides of the blister pack holder were carefully eliminated by sanding them to a level surface.

The entire center section of the blister pack holder required a replacement piece as the pill holes did not line up with the design reference blister pack. This was done by carefully cutting out the center of the blister pack holder removing all of the original holes while leaving the remaining external stiffening rim untouched. The design reference blister pack was measured to determine the needed spacing between pill blisters and the diameter of the holes for the new pill holder. A new center piece was then fabricated from a spare piece of .031 inch thick FR4 (circuit board material). This piece was then bonded to the bottom of the blister pack holder with epoxy to provide the completed blister pack holder. Figure 7 shows the comparison

between the original and the final modified blister pack holder, and Figure 8 shows the assembly of the of the modified blister pack holder.



**Figure 7.** Modifications to Blister Pack Holder Center Piece



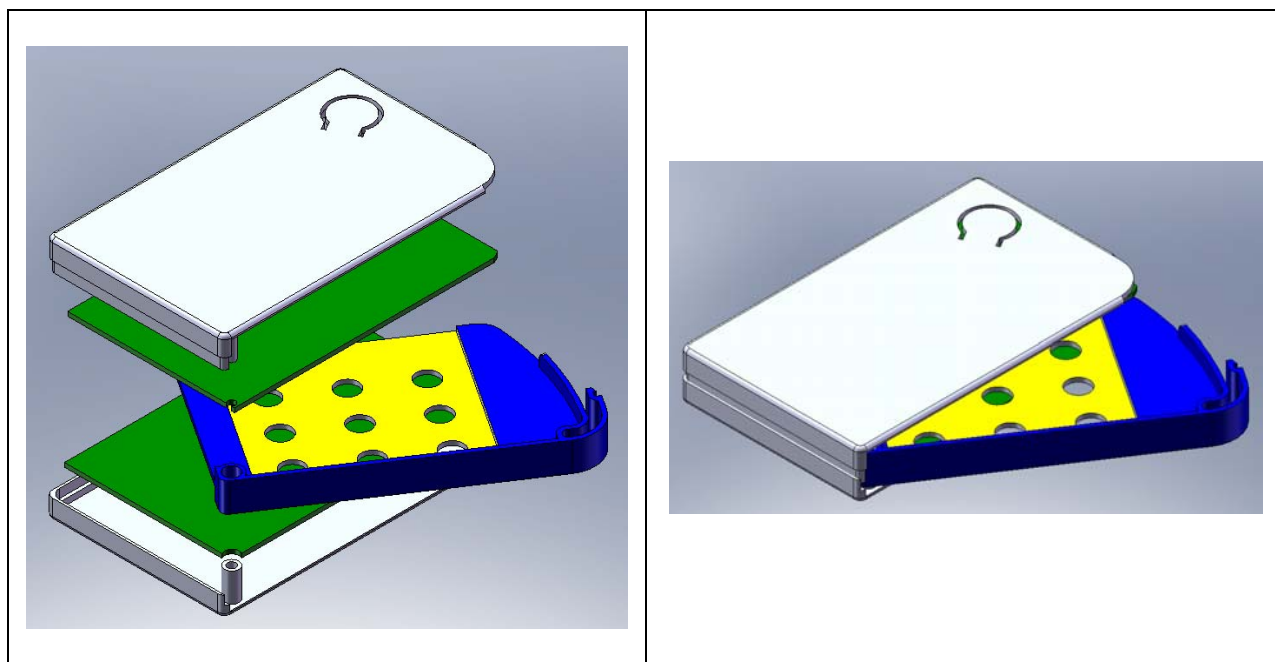
**Figure 8.** Assembly of Blister Pack Holder

The original clearance between the pill holder and the top of the case was 0.18 inches, but the blister pack and pills combine to be this same thickness leaving no clearance between the pills and the top of the case. This causes a problem as there is no room for electronic



components to be added to the top of the case. The newly fabricated blister pack hole piece was bonded to the bottom of holder to recess the position of the blister pack providing a 0.07 inch clearance between the top of the blister pack and top of the case. The tradeoff is a 0.031 inch reduction of the clearance of the bottom to the holder due to the thickness of the FR4.

The device was designed to utilize thin 0.02 inch PCB's to allow the pillbox to be fully assembled and maintain the same original external dimensions. Unfortunately, ordering such a board is a specialized expensive task requiring increased lead time to receive the PCB. In the interest of time and given that it was a proof of concept device, a standard 0.062 inch PCB was ordered and the case was slightly changed to accommodate the increased thickness. The boards were made so that they would slide into place around the boss pivot that supports the center piece. When the pillbox is fully assembled, the center piece is to be flush with the PCB's, not the inside of the case effectively increasing the thickness of the pillbox by 0.124 inches (2 board thicknesses). This prevents the top and bottom pieces of the pillbox from fully snapping closed as shown in Figure 9 below.



**Figure 9.** Modified Case Assembly with PCBs

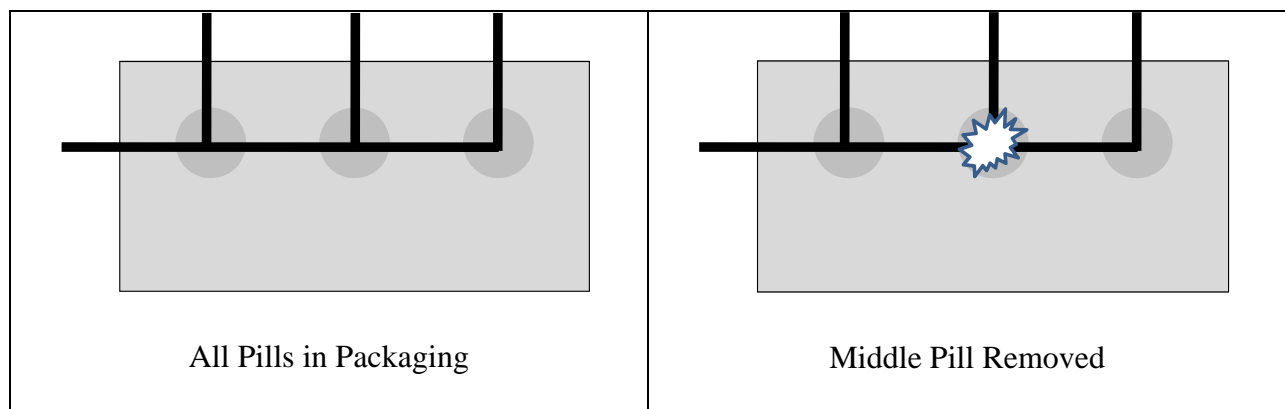
## **3.2 PILL DETECTION TECHNIQUES**

The pill detection method must be able to reliably determine which pills have been removed from the blister pack. Three different innovative techniques were conceived to provide this functionality, and the benefits and disadvantages of each are discussed below with the best option being chosen.

### **3.2.1 Push Through Tracks**

The first concept considered the use of custom blister packs in which the foil area was replaced with a back film that would contain electrical tracks across the blister areas. When pushing a pill through the film, the electrical track would be broken, which is analogous to opening a switch.

By electrically checking the status of these tracks, you can determine when a track has been broken. The method utilized for reading matrix keypads inspired this approach and a track layout was considered to match this type of system as shown in Figure 10. The matrix keypad method usually consists of a grid of rows and columns in which there is no connection and each button of the keypad provides a link between the two. By periodically driving the rows with a voltage, and monitoring all of columns, it is quite easy to detect a change in state when the connection is made, and determine which button was pressed, or in this case, which pill was removed.

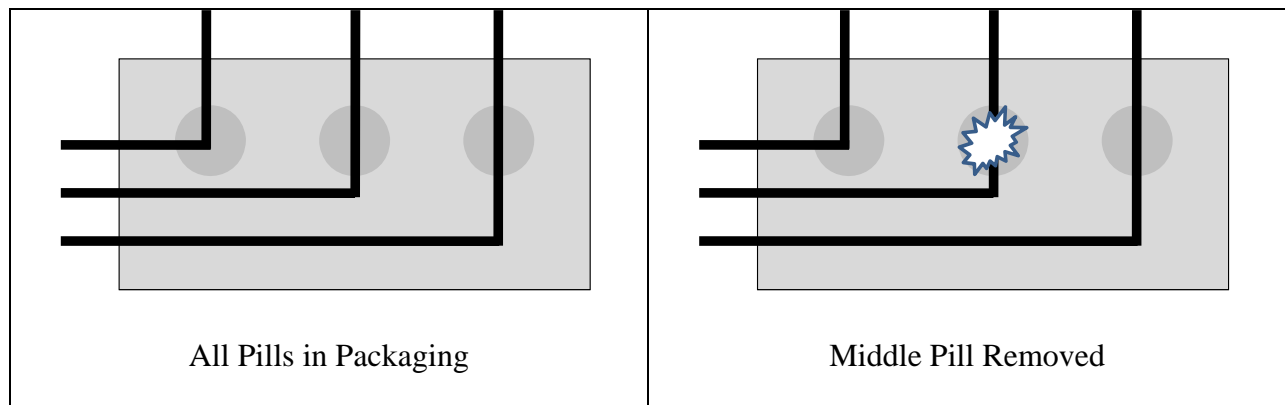


**Figure 10.** Push Through Tracks Method Emulating Matrix Keypad, Bottom of Blister Pack

Unfortunately, this method is not directly applicable because the matrix keypad consists of momentary normally open switches. The normally closed nature of the blister pack tracks would become permanently open once the track is broken causing problems. The permanently broken tracks can lead to pill locations being completely electrically isolated and pill locations unable to be checked by the matrix method. Figure 10 illustrates this problem as the last pill on the right becomes electrically isolated after the middle pill has been removed.

The solution to this problem is to have individual inputs and outputs for each blister location as shown in Figure 11. This unfortunately makes the fabrication of the tracks for the

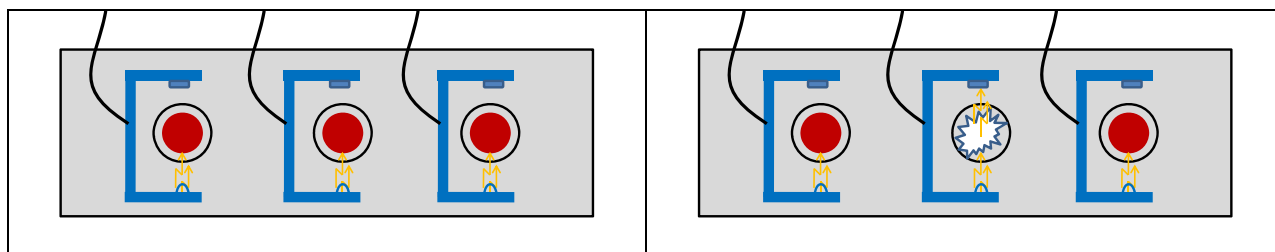
film complicated and expensive as many more traces are required in a small space. The connections to such a layout would also be complicated. These disadvantages make this solution unattractive as it becomes overly complex and difficult to adopt.



**Figure 11.** Push Through Tracks Method with Individual Traces for Each Pill, Bottom of Blister Pack

### 3.2.2 Infrared (IR) Interrupters

IR interrupters are devices which combine an IR emitter and receiver. The output of the device depends if an obstruction is blocking the line of sight between the emitter and receiver. These devices could be placed and oriented horizontally such that the IR beam would be blocked by a pill within a blister. Once the pill is removed, the beam would pass through the empty blister providing a pill detection method as shown in Figure 12.



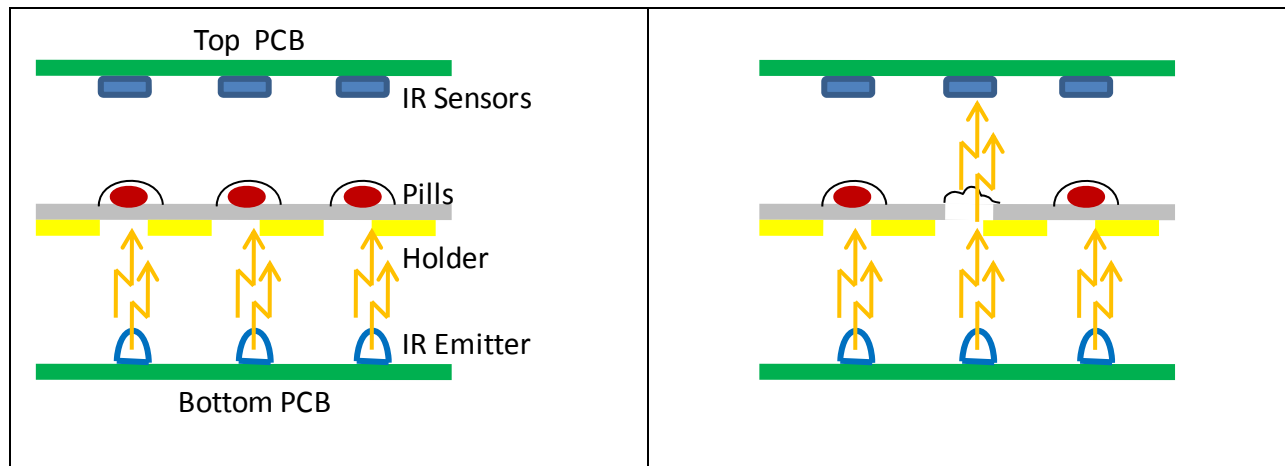
**Figure 12.** IR Interrupter Method, Top of Blister Pack

Unfortunately, this arrangement has many issues, the first of which is exposing the complex nature of the pill detection to the patient or end user as the interrupters would be placed on top of the blister pack. This also causes complexities for the pharmacies as refilling would become a more difficult task involving aligning the interrupters properly. The size of the interrupters themselves causes problems as they define the minimum space between blisters in a pack making the compact design difficult. The method could also be unreliable as the pills can move around within the blister possibly enough to allow a line of sight connection between the emitter and receiver causing false positives. It is also assumed that the pill is opaque enough to block IR, which is not guaranteed. For these reasons, the IR interrupter solution is also not an ideal design and was not utilized.

### **3.2.3 IR Sandwiched Array**

By combining the best qualities of the previous two solutions, the disadvantages of both are eliminated. An array of IR emitters and IR receivers/sensors are arranged vertically with a line of sight connection through the pill openings of the blister pack holder. This line of sight beam is blocked when the holder is loaded with a new blister pack by not only the pills, but also the film layer eliminating the reliability concerns of the IR interrupter method. Once the film layer is broken and the pill removed, the IR emitter radiation can reach the sensor creating a pill

detection method as shown in Figure 13. The disadvantage of such a method is the small possibility of the broken film falling back into place over the empty blister blocking some of the light. The affect of such an event is minimal as high sensitivity IR sensors will trigger for any IR light detected and a broken film cannot entirely block the light from the top PCB layer. Future studies should however be performed to determine the probability of errors for such an event.



**Figure 13.** IR Matrix Method, Cross Sectional View

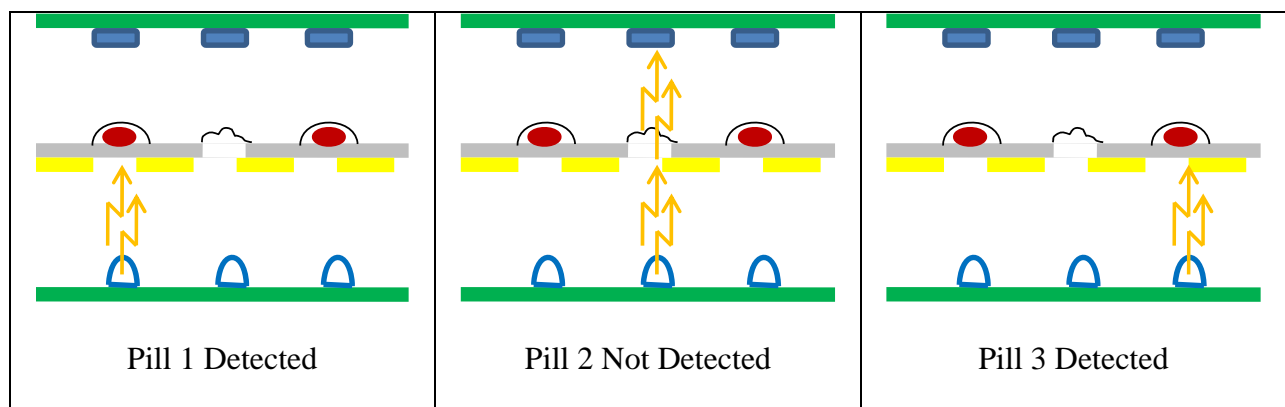
With these arrays attached to the inside of the top and bottom of the pillbox case, the blister pack can be checked whenever the pillbox is closed. The advantage of this arrangement is that the medication monitoring hardware is hidden and protected from the user and the pillbox appears no different than the original casing. The use of two PCB's, one for the top and one for the bottom, also provides increased board area making it easier to distribute the required Integrated Circuits (IC's) for the pillbox. The tradeoff is the slight design complexity added by the alignment of the IR arrays and the connection method between the two PCB's on which the IR emitters and sensors are attached. The Solidworks model provided essential information and dimensions required to solve these additional complexities.

### 3.3 PILL DETECTION DESIGN

The implementation of the IR sandwiched array requires the choice of appropriate components that fit the space constraints and provide adequate performance. The functionality of each of the components was tested in a mockup system to verify their operation and to determine the required values for supporting passive components.

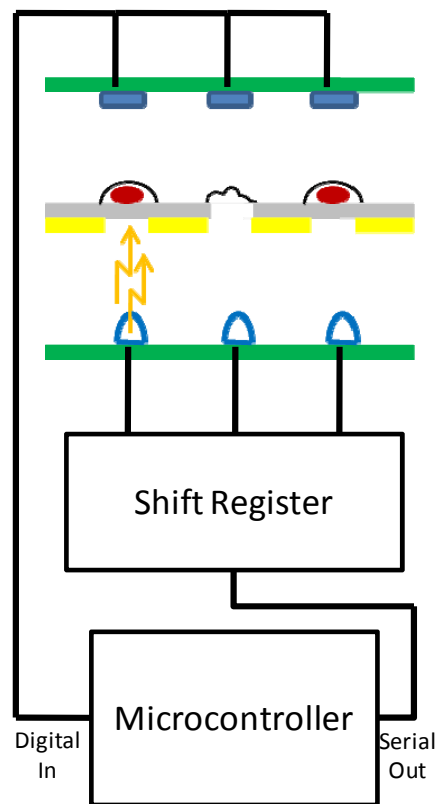
#### 3.3.1 Operation

In order to minimize the number of traces and required input and output lines to the microcontroller the following strategy is followed for pill detection. A single digital input line to the microcontroller monitors the net output of the phototransistors so that any light detected by any of the sensors will trigger the output. At any time, only one IR emitter will be activated. If a sensor detects an IR emission, the light path is not blocked and the pill associated with the activated IR emitter has been removed. Figure 14 illustrates this process with the middle pill removed from the blister pack.



**Figure 14.** Pill Detection Operational Example

In order to minimize the number of outputs from the microcontroller controlling the IR emitters, a shift register system is utilized in which each IR Emitter is sourced by an output of the shift register. A single IR emitter is activated by shifting a single “high” bit through the shift register. This process only requires two digital output lines from the microcontroller for serial communication to the shift register. Figure 15 shows the functional connections between the pill detection method and the supporting microcontroller hardware.



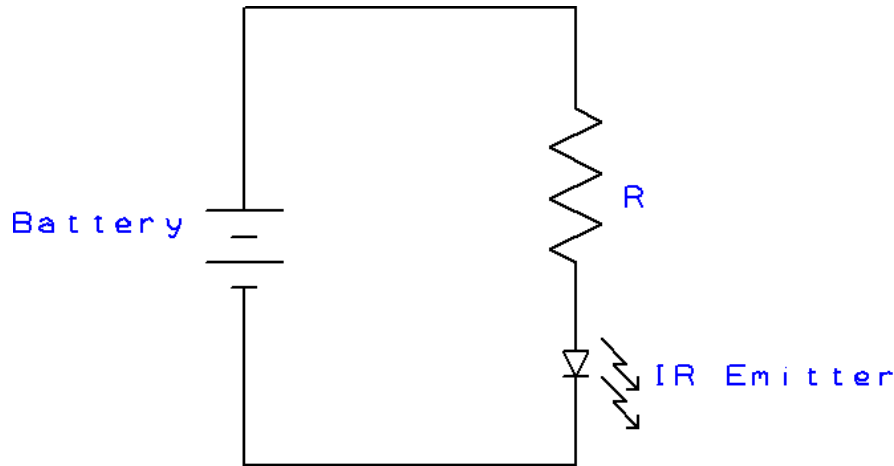
**Figure 15.** Pill Detection Functional Hardware Connections

### 3.3.2 IR Emitters

IR emitters are electronically equivalent to diodes with the intensity of the radiation directly proportional to the amount of current flowing through the device. The current is controlled



utilizing a current limiting resistor in series with the device as shown in Figure 16. The amount of current is determined by utilizing Equation 1 below with the rated limits of the IR emitter provided by the data sheet [15].



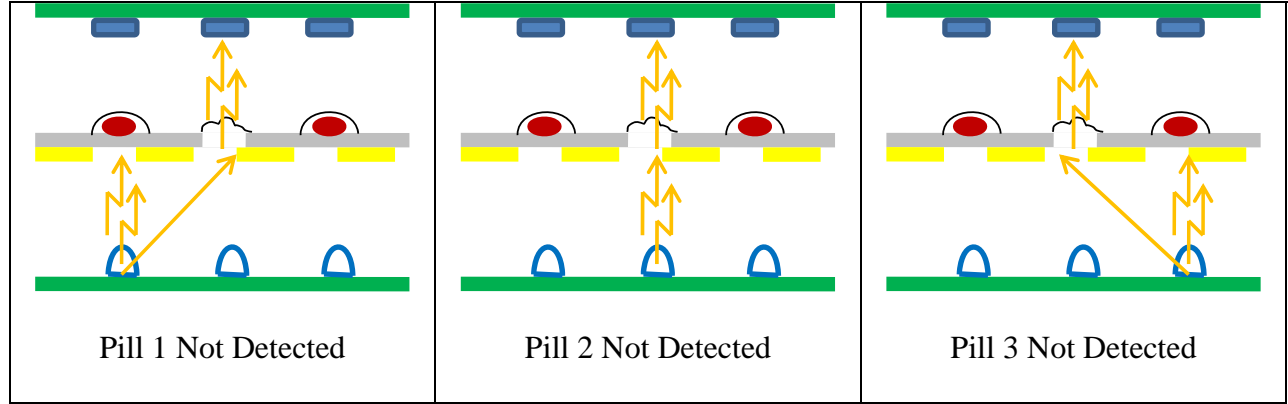
**Figure 16.** IR Emitter Current Limiting

$$i = \frac{V_{Battery} - V_{Forward}}{R}$$

**Equation 1.** Current Calculation for IR Emmitter

These devices are typically used for remote control applications in which the intensity of the radiation must be strong enough to transmit encoded information over a distance on the order of yards. As such, the rated 20mA forward current in the data sheets match these requirements. This application is much different as the device only needs to create light with an intensity to travel an inch. As such, much less current and power is required extending the battery life of the device and easing the design constraints for the current supply source. The value of the resistor, however, needed to be experimentally determined with a mockup system to determine if the entire system could operate with this lower IR Emitter Current.

The selection of the IR Emitter depended greatly on the half angle power of the device. A narrow beam angle was desired to ensure that the radiation from one IR Emitter would not shine into the pill opening of a neighboring pill location as shown in Figure 17. Such a situation would cause false pill readings.



**Figure 17.** Pill Detection Beam Angle Error Example

Given the diameter of the pill holder holes (0.33 inches) and the distance of the IR Emitter to the pill holder (0.149 inches), the propagation angle to the aperture can be calculated with Equation 2.

$$\theta = \arctan\left(\frac{\text{radius of aperture}}{\text{distance to aperture}}\right)$$

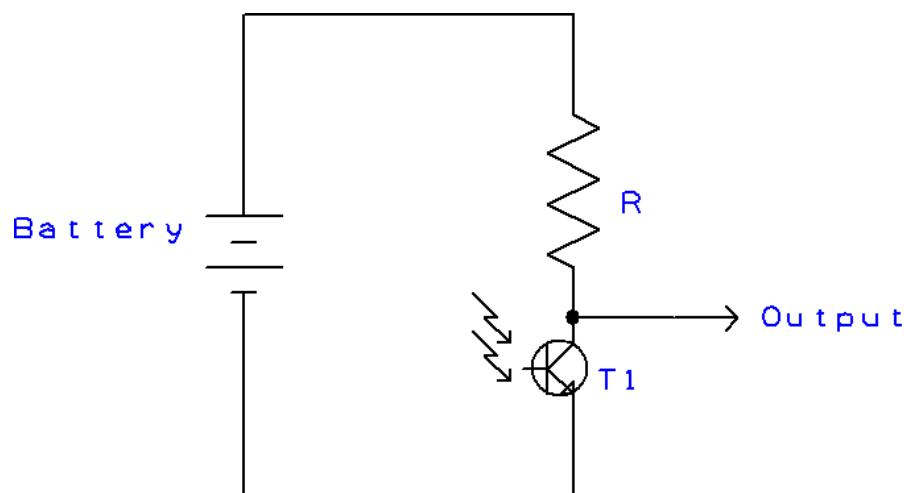
**Equation 2.** Propagation Angle from IR Emitter to Pill Holder Hole

The resulting propagation angle is approximately 48° which requires the IR intensity to be reduced for propagation angles larger than this limit. Surface mount IR Emitters are produced in two types of packages: a thin broad propagation angle device, and a thicker narrow propagation angle device with an integrated lens. It was determined that the thickness trade off was acceptable for the increased control of the beam directionality. The device chosen is less

than 0.090 inches thick and is less than 20% intense for propagation angles greater than  $30^\circ$  [15]. The peak emission wavelength is 940nm, which must be matched to the IR Sensor chosen.

### 3.3.3 IR Sensors

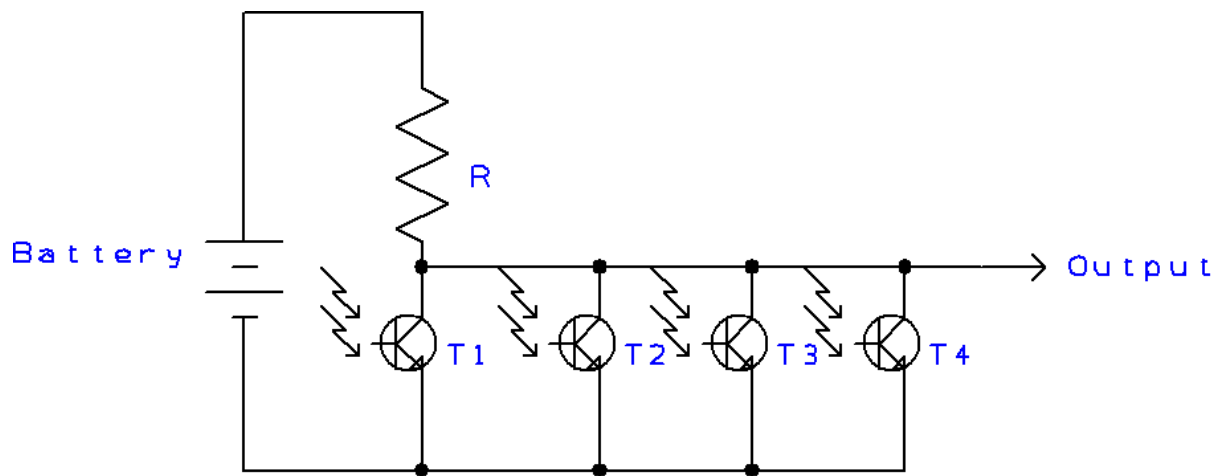
IR sensors are available either as photo transistors or photo diodes. Photo diodes have a linear response with respect to the intensity of IR they are exposed to. Unfortunately, they require the use of a current amplifier and tend to be relatively expensive. Photo transistors on the other hand are relatively inexpensive, but are not as linear as photo diodes and do not need a current amplifier as they can be configured as a high gain common emitter amplifier as shown in Figure 18. As this application does not require a linear response, the photo transistor is the ideal component selection. The OP521 photo transistor was chosen due to its small size, and matching spectral sensitivity with the peak occurring between 900-950nm [17].



**Figure 18.** Common Emitter Amplifier Configuration

By selecting a large collector resistor, the output voltage will saturate when the transistor is exposed to any IR radiation of the proper wavelength. This saturation will cause the output to

be driven to ground or logic 0 when any of the sensors are exposed to light as the voltage across the collector resistor will dissipate all of the voltage. The outputs of all of the sensors are combined by using one common collector resistor as shown in Figure 19. The resulting voltage drop across the resistor is a function of the sum of the current draw from all of the phototransistors. The value of the collector resistor was determined experimentally using a mockup system.



**Figure 19.** Photo Transistor Superimposed Network Connection

### 3.3.4 Component Testing

The IR Emitters are powered by the outputs of a shift register as shown in Figure 15. The current sourcing ability of the shift register's output is important as it must be able to provide adequate current and voltage to the IR Emitter to ensure consistent operation. If the current capacity of the shift register output is exceeded, the voltage level at the output will drop causing problems as the IR Emitter will not be supplied with the voltage and current for which it was configured.

The 74LV4094 shift register was chosen for this application due to its compact size and ability to operate with voltages as low as 1V. Unfortunately, the data sheet does not provide a clear parameter for the amount of current one of its outputs can source requiring that experimental tests be performed to determine its operational limits. Breadboard testing of the shift register showed that for 3V power supply, the 74LV4094 could source 20mA of current but only at 2.35V. For current demands below 4mA, the output voltage of the shift register nearly matched the 3V of the power supply providing a consistent power source for the IR Emitters.

The IR Emitters' current limiting resistors were chosen to limit demanded current to values below 4mA at 3V as constrained by the shift register. A resistor value of 470 ohms was calculated to meet these requirements, and breadboard testing verified that the IR Emitter could successfully be powered by the shift register outputs in this configuration. IR Emitter operation was viewed utilizing a digital camera and its Liquid Crystal Display (LCD) preview functionality as IR light is invisible to the naked eye. A digital camera's sensor can detect IR light and would show the IR emitters giving off blue light if they were active.

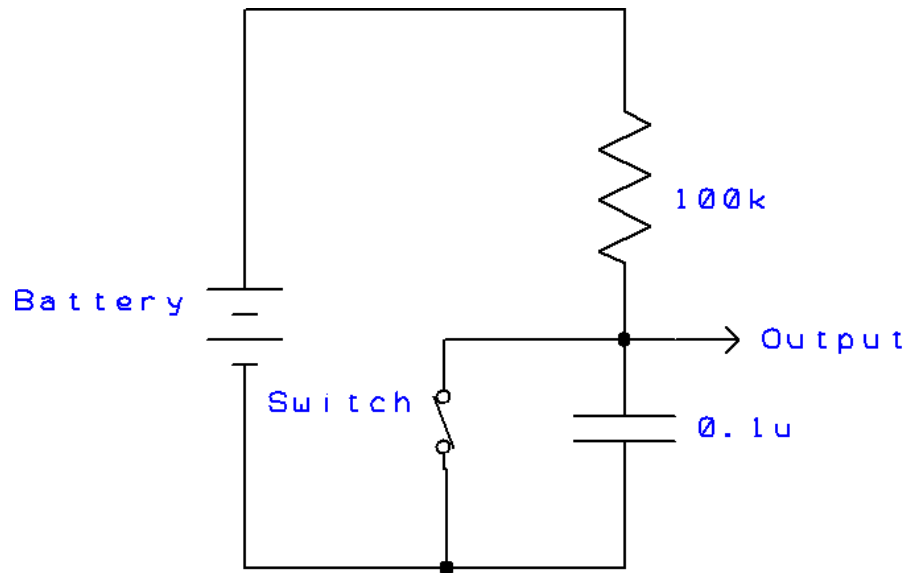
The photo transistor collector resistor value was determined experimentally by varying the resistor values and monitoring the resulting output voltage when exposed to an active IR emitter. A decade box was utilized to provide adjustable resistances during testing. As florescent lights emit IR radiation in the range being detected by the photo transistor, all testing was performed in the dark with the lab lighting turned off. The results of the testing revealed that a 100k ohm resistor was sufficient to provide logic level voltage outputs even for the lowest current IR emitter conditions (dimkest IR illumination). A larger value of 470k ohms was chosen for the final design in order to provide a design margin with increased gain to ensure voltage saturation for the needed logic level outputs.

### 3.3.5 Event Trigger

The Pill Detection method should only be utilized when it is known that a pill may have been removed in order to extend battery life. Given the case structure and the operation of the pill detection system, it is known that a pill can only be removed when the case is opened. By detecting when the case is opened and closed, the system can efficiently check the pill states as soon as the case is closed.

A small surface mounted switch provided the perfect method to check for the case opening and closing events. A small lever on the switch is depressed by the outer rim of the blister pack holder providing a logic output to the microcontroller signaling when the pill detection cycle should begin. All switches unfortunately have inherent switch bounce which can cause multiple logic transitions for a single switch event. In order to avoid this jitter, a de-bounce circuit was implemented.

There are many methods to de-bounce a switch, but the simplest hardware implementation is a passive low pass filter, which only requires a resistor and capacitor. This implementation eliminates the high frequency transients that are inherent in switch bounce leaving a smooth transition for a switch event. A latch de-bounce implementation was also a consideration, but was not utilized due to the additional IC required. Figure 20 below illustrates the layout of the switch de-bounce system used for the pillbox.



**Figure 20.** Switch De-bounce Circuit

### 3.4 MICROCONTROLLER

The microcontroller for the project manages the all of the functionality needed by the pillbox including pill detection, power management, and communication. These tasks are interconnected requiring both accurate timing and proper software organization to operate successfully. The capabilities of the chosen microcontroller and its impacts on the software organization are discussed in following sections. The overall software organization is then described following the program flow of the device once it is powered. The details of each functional block within the software are then shown with logical flow diagrams describing their operation.

### 3.4.1 Selection

The project requires a low power microcontroller that runs off of low voltage power supplies with power management capabilities in order to maximize battery life. The communication and Real Time Clock (RTC) functions depend on accurate timing, which is provided by timer hardware and interrupt routines. Many microcontrollers are available that fulfill these requirements, but TI's MSP430 line is specifically designed for compact low power applications, and this product line has already proven to be more than capable when used in active RFID applications [21]. Microchip's PIC devices were also considered, but the comparable low voltage varieties for this application were not available at the time of ordering, and most specifications favored the MSP430 including a 16 bit architecture in a small form factor.

The MSP430F2132TPWR family variant was chosen for the project for its small package size and more than adequate specifications including 8K of program memory, 512 RAM, and 24 digital I/O pins [16]. Two 16 bit timers (Timer0 and Timer1) and associated capture and compare registers (A0-A3) provide all of the critical timing functionality needed. Each Timer has capture and compare registers associated with them which provide the interrupt capability. The microcontroller is designed to operate with voltages as low as 1.8V maximizing the time the device can run on a battery. Less than 5mA of current is required by the MSP430 when in active mode, and only 0.2uA when the device is in sleep mode. An internal Digital Controlled Oscillator (DCO) can be configured to drive the main system clock (MCLK) up to 6MHz at 1.8V providing more than enough processing power for the pillbox. High speed crystals could also be used for the main clock, but at the cost of increased power consumption and component count.



### 3.4.2 Limitations

The only limited functionality discovered with the MSP430 is the inability to reconfigure the interrupt priorities of each interrupt source. There are 32 interrupts available, each with a predefined priority value. Higher priority events are defined by higher priority values (ie. interrupt priority 32 is the highest priority interrupt). This can cause problems especially with timing interrupts as each capture and compare register has different and sometimes unique clock source inputs available. This can constrain which timer can be used and dictates which interrupt priority is assigned to a given functionality, which may conflict with the overall system design.

The pillbox prototype utilizes interrupts heavily in its operation for power saving operation and accurate timing. Table 1 shows the interrupts used for the pillbox and the interrupt priorities of each.

**Table 1.** Priority of Interrupts Used for Pillbox

Function	MSP430 Name	Priority
Power up	Power up	31
RTC	Timer1_A0	29
RF Comm	Timer0_A0	25
Clock Reg	Timer0_A2	24
Switch	P2	19

The Power up interrupt is the routine that executes when the microcontroller is first powered up and provides the initial configuration for the processor. The event triggered switch discussed in Section 3.3.5 is tied to a Port 2 (P2) digital input to provide an external interrupt source to wake the processor when it is in the power saving sleep mode. The three timer interrupts provide the accurate timing required to provide real time clock (RTC), MCLK regulation (Clock Reg), and RF communication (RF Comm) functionality. The clock regulation

provides a control method to ensure that the main clock is operating at the expected frequency improving the system's overall timing accuracy.

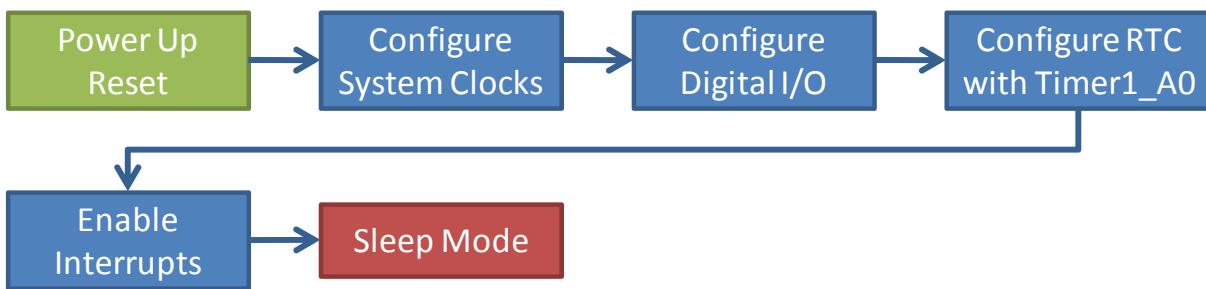
The RF Communication function would ideally be the highest priority of the timer interrupts in order to guarantee consistent communication timing, but other design constraints force the priority layout given above where the RTC has more priority. This is due to the fixed priorities of the timers, and the clock regulation functionality requiring the use of a unique clock source available to the Timer0\_A2 interrupt.

The RTC requires a dedicated timer that cannot be reconfigured to ensure the RTC accuracy. As Timer0 must be configurable for clock regulation, Timer1 and the Watchdog Timer are left to be used. Timer1 was chosen because it is much more flexible and allows for the RTC interrupt to be triggered every minute providing clock updates. The watchdog timer's structure would have required the use of 4 second RTC interrupt intervals to provide one minute resolution desired (due to limited configuration options). This would slightly reduce the overall battery life as the device would wake up 15 times more often than the Timer1 configuration.

As the Timer1 configuration cannot be modified due to the RTC requirements, the RF Communication functionality must use Timer0. This does not disrupt the Clock Regulation functionality as it completes its execution before the RF Communication functionality can begin (software organization). These timer constraints force a non ideal interrupt priority as the RTC has priority over the RF Communication. The operational risk of this arrangement is minimal however as the RTC is a very short subroutine that runs only once a minute, which greatly reduces the chances that it would disrupt an RF communication interrupt.

### 3.4.3 Initialization and Configuration

When powering up, the microcontroller initiates preconfigured clock settings and enables the proper digital I/O needed for the pillbox operation. The device then configures the periodic interrupt for the RTC and enables all interrupts before entering sleep mode to conserve power. The microcontroller only wakes up when an interrupt occurs, which can be triggered by either the RTC timer or the event triggered switch. Figure 21 shows the progression of the initialization and configuration operations for the pillbox.



**Figure 21** .Initialization Sequence

The configuration parameters and register values were determined by information from both the MSP430 datasheet [16] and user's guide [11]. The DCO can be configured by setting both a DCO value and a modulation value. The DCO value defines a baseline oscillator frequency providing course adjustment of the MCLK frequency. The modulation setting mixes this baseline oscillator with the next highest DCO frequency setting to provide fine adjustment of frequency settings between the DCO settings.

The system clock is configured for a DCO value of 4 with no modulation ideally providing a theoretical 6MHz MCLK to the processor. This MCLK frequency was chosen as it is the maximum the microcontroller can provide at the 1.8V low voltage level. This DCO

frequency will vary between chips and also their relative temperature, which causes accuracy problems. Clock regulation functionality was included with the design to address this concern and to provide an accurate MCLK.

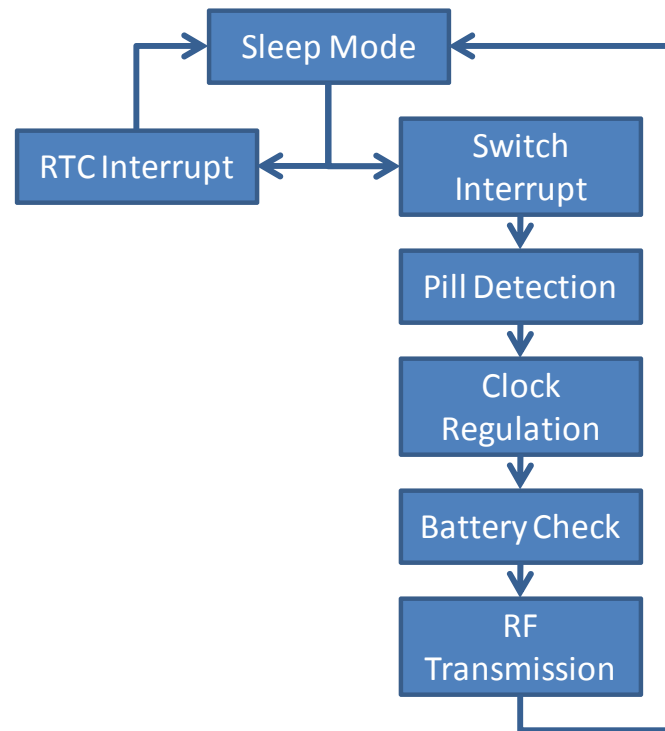
A watch crystal providing a low frequency 32.768kHz reference is utilized by the microcontroller to provide an accurate time base for both the RTC and clock regulation functions. This low frequency is made slower by dividing it by 8 before being used by any of the timer routines as the Auxiliary Clock (ACLK) resulting in an effective frequency of 4096kHz. This small crystal typically requires small bypass capacitors to enable the oscillation, but the MSP430 is designed with internal capacitors for this functionality eliminating the need for external capacitors to be added to the board layout reducing the overall component count.

The digital I/O is configured to provide the interfaces required for both the pill detection and the RF communication functionality. The event trigger switch that detects the pillbox closures is connected to a port 2 digital input pin with edge triggered interrupts configured. All of the interrupts are then enabled on the microcontroller and the device power management settings are initialized. The sleep mode is a power setting in which only the internal microcontroller hardware (including timers) and slow ACLK are running with both the processor and MCLK disabled. The microcontroller only wakes up the processor to service interrupt routines, which were previously configured.

#### **3.4.4 Overview**

Once initialized, the software has two possible interrupts to wake it from the sleep state. The first is the simple RTC algorithm which executes each minute and only wakes the processor up to update the time. The second is the switch interrupt which is more complicated as it is

responsible for multiple functions when the pillbox is closed. Figure 22 shows an overview of the operational functionality of the pillbox.



**Figure 22.** Software Overview

The switch interrupt enables the processing of the pill detection algorithm, clock regulation, battery check and RF transmission functions. The switch interrupt service routine wakes the processor and disables the sleep mode ensuring that the MCLK will remain operational for all of the following functional blocks. This is essential as the MCLK must be active for both the clock regulation and RF transmission functions. The pill detection function then updates the status of all the pills within the case and stores the results. MCLK frequency regulation begins by configuring the proper timer and interrupt settings. This functionality ensures that the DCO is configured to provide a consistent 6MHz for the MCLK. This is needed as the RF transmission protocol requires an accurate time base (MCLK) to provide precision

pulse widths and bit rates to match the RFID standards. The high frequency MCLK must be utilized for this functionality in order to provide the needed time resolution given the microsecond scale the communication standard requires [22]. The battery voltage is then measured and stored, and the current state of the pillbox is transmitted wirelessly. Once the RF transmission finishes, sleep mode is re-enabled and the device returns to sleep mode to save power.

### 3.4.5 Real Time Clock

In order to provide a time stamp for pill removals, a real time clock must be implemented in the device. Assembly example code was available from the TI's website that implemented a RTC with only a 32.768kHz crystal to provide the time base and a timer to count the cycles of the crystal [10]. As the project software was written completely in C, the assembly example was analyzed and rewritten in C. The code was also modified to provide an interrupt interval of 1 minute instead of 1 second. This was done to reduce the number of times the microcontroller would have to wake up to process the real time clock saving power and increasing battery life.

Since the ACLK is already operating at 4096kHz, Equation 3 below shows the calculation of the timer parameters used to provide the minute resolution functionality.

$$60s = x \left( \frac{y}{4096} \right)$$

**Equation 3.** Calculation of Timer Configuration Values for RTC

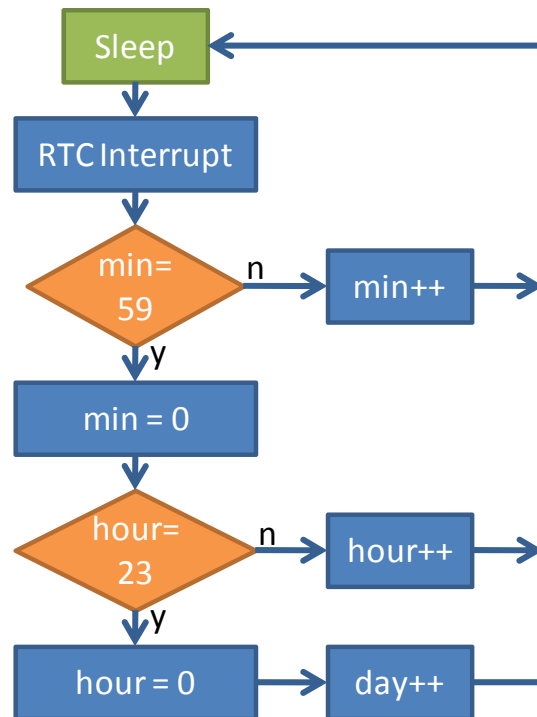
$x$  is the input divider of the timer, which can be chosen to be 1, 2, 4, or 8; and  $y$  is the 16 bit value that causes the timer to reset and generate an interrupt. Table 2 provides the solutions of  $y$  for the possible values of  $x$ . Only the final two entries are possible however since  $y$  is

limited to 16 bit values. The highlighted entry is the combination used for the RTC configuration.

**Table 2.** Solutions to the Timer Configuration Equation

x	y
1	245759
2	122879
4	61439
8	30719

Figure 23 provides the functional details of the RTC interrupt routine. Each interrupt increments the min variable and checks to see if the hour and day variables should also be incremented.



**Figure 23.** Real Time Clock Functionality

It was decided that the pillbox itself would not store the exact calendar date and time and would instead act as a stopwatch counting the relative time since the pillbox was refilled. This reduces the complexity that would have been required to implement monthly corrections and leap years in the real time clock, and reduced the slight inaccuracies with the crystal that can lead to time drift errors. When the pillbox is refilled at the pharmacy, the receiving station at the pharmacy should be able to detect a refill event for the pillbox, and store the current date and time of the refill event. The pillbox itself will reset its internal clock values during the refill process. The time stamps provided by the pillbox are then offsets to be added to the stored refill time stamp to provide the calendar dates and times for all pill removals.

### **3.4.6 Pill Detection**

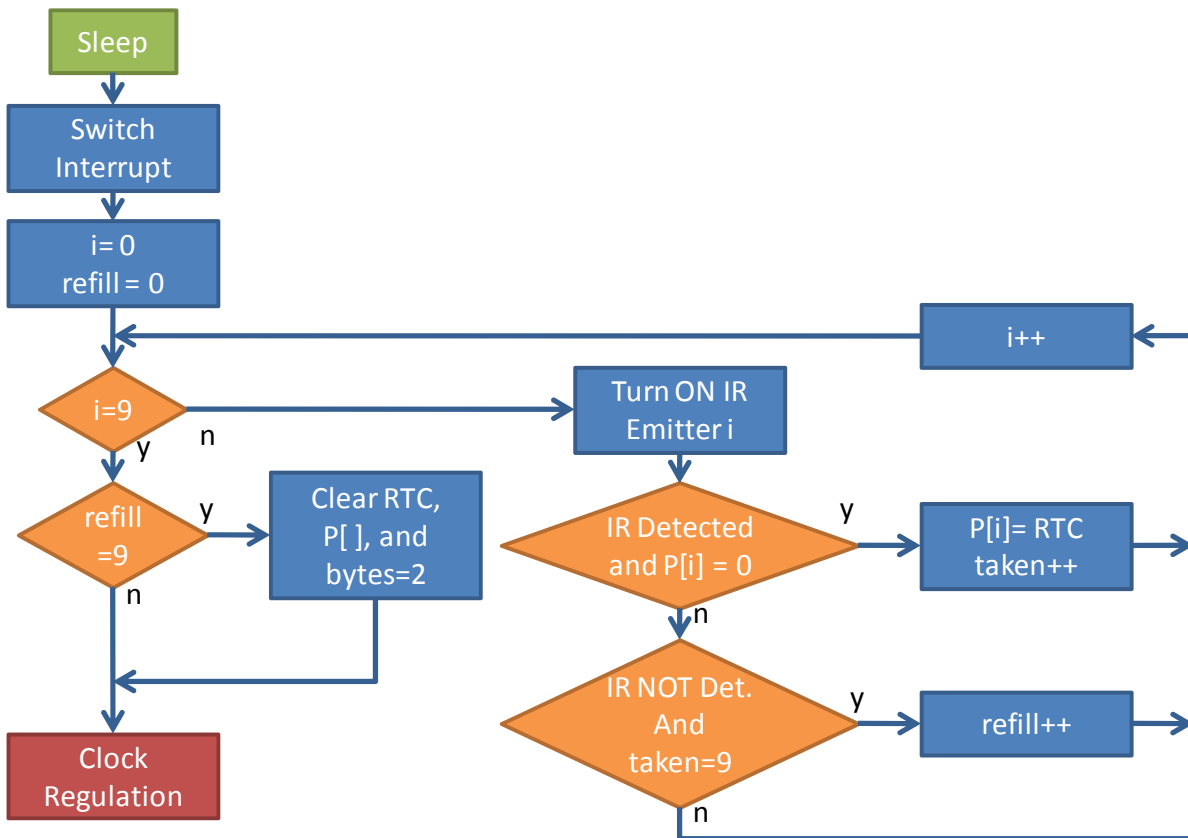
The pill detection algorithm is executed as soon as the switch interrupt is triggered. The variable  $i$  tracks the pill number currently being checked, and the variable *refill* determines if all of the pills have gone from empty to full indicating a new blister pack has been inserted into the pillbox. The array  $P$  stores the status and time stamps of all of the pills.

The algorithm operates by turning on one IR Emitter at a time and checking to see if the sensors detect IR. If they do, the pill has been removed and the appropriate  $P$  array index is updated with the current RTC value and the *taken* variable is incremented. Once a pill has been classified as removed, the device will not change its record for that pill until a refill condition is detected. This is to ensure the time stamp record of the pills will be accurate and unchanged until someone refills the pillbox.

Once the taken variable has reached 9, indicating that all of the pills have been removed, the device checks for the refill condition, which occurs when a new blister pack is inserted.



When the refill condition is met, the RTC variables and *P* array are reset and the *bytes* variable is changed to two. The *bytes* variable controls how many bytes of data the device transmits as described in section 3.4.9.2. A receiving station detecting a short data packet will know that the pillbox has just reset itself and can take appropriate actions. Figure 24 shows the logical structure of the pill detection algorithm.



**Figure 24.** Pill Detection Functional Diagram

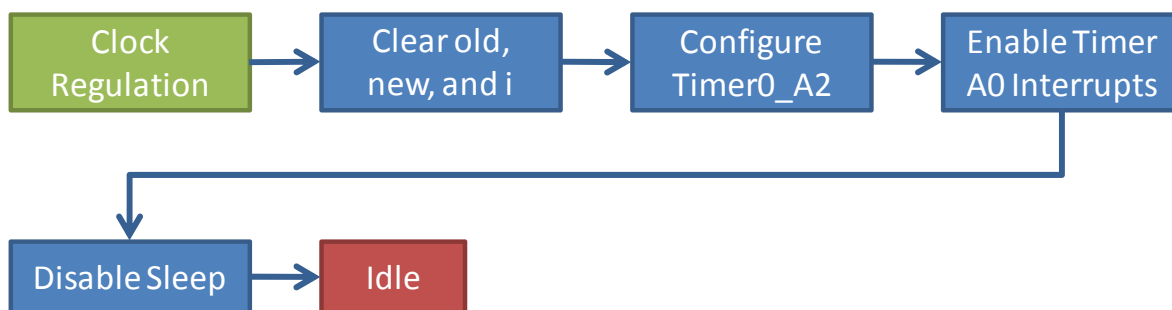
### 3.4.7 Clock Regulation

The DCO has multiple levels of control available to modify the frequency. By incrementing or decrementing the DCO register (contains both the DCO value and modulation value), the

resulting frequency can also be increased or decreased. Example assembly code provided by TI's website provided two control methods to regulate the DCO frequency using a timer and a 32.768kHz crystal [9]. As before, the code was examined and then written in C for the pillbox. The first method was a proportional control loop which had a quick response, but required tuning. This is a good option for changing frequencies in large increments, but not needed for this application as the target frequency is already close to the originally configured DCO values. The second method is known as a sliding window method in which the DCO register is only incremented or decremented by a single value providing a fine tuning.

The MCLK frequency error was determined by comparing the number of clock pulses recorded within the reference ACLK cycle. Depending on the sign of the resulting error, the DCO register is incremented and decremented to minimize the error. Once the error is eliminated for 3 ACLK interrupt cycles, the MCLK is considered regulated and accurate, and the remaining functions could be executed.

This control algorithm is dependent on counting and accurate timing events requiring the use of interrupts. Figure 25 shows the configuration of the capture compare timer interrupt for MCLK regulation. The end state of the configuration is idle as the MCLK is still active and the processor is waiting for an interrupt event to begin execution.



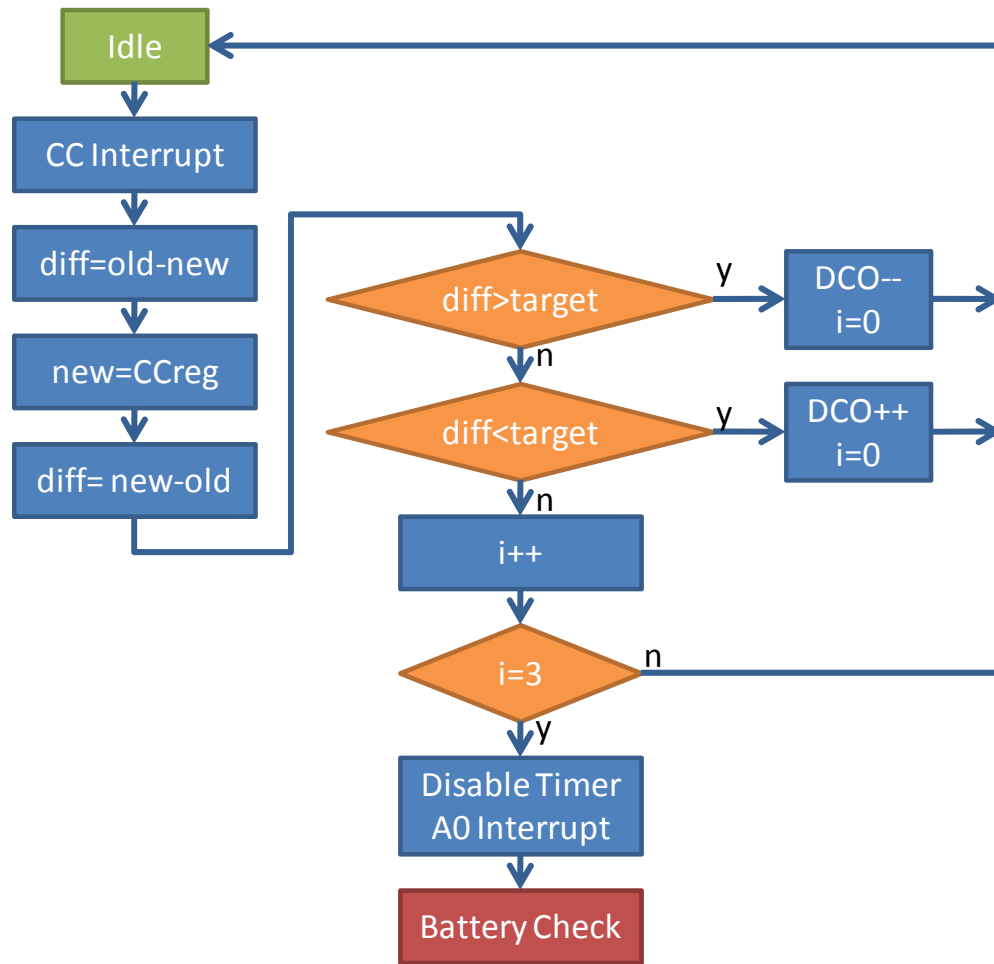
**Figure 25.** Clock Regulation Functional Diagram

The interrupt operates by triggering when a rising edge event is detected from the ACLK and storing the timer (counter) value, which is driven by the MCLK. By taking the difference between two capture compare interrupt events, the number of MCLK cycles for an edge transition of the ACLK can be calculated. This difference can then be compared to the target value of 1464 cycles which is calculated from Equation 4.

$$1464 \text{ Cycles (Timer Target Value)} = \frac{6000000\text{Hz (MCLK)}}{32768\text{Hz (ACLK)}}$$

**Equation 4.** Target # of MCLK Cycles per ACLK Cycle

The *old* variable is the timer value from the previous ACLK interrupt event, and *new* variable is the timer value from the current ACLK interrupt event. The variable *i* tracks the number of consecutive errorless interrupt events. Figure 26 shows the detailed operation of the capture compare interrupt. Once the MCLK frequency has been regulated, the battery check function is executed.



**Figure 26.** DCO Control Interrupt

### 3.4.8 Battery Monitoring

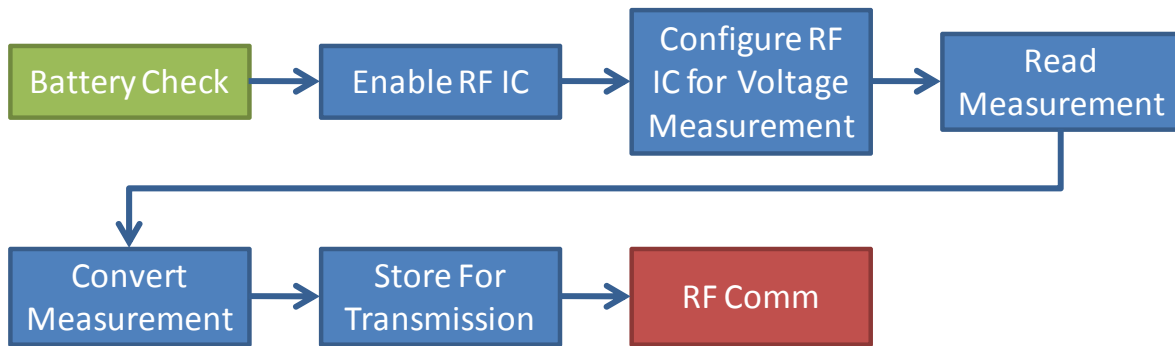
The battery voltage can be measured by two methods. The first method was provided by an online article and utilizes an internal MSP430 voltage reference to compare to a voltage divided supply rail value using one of the comparator inputs of the microcontroller [13]. Once the threshold of the internal voltage reference is passed, it is known that the supply voltage has dropped below a set value which can indicate the need for a battery replacement.

The second method utilizes the functionality included with the RF Integrated Circuit (IC) used for the RF communication for the device. The ADF7020 can be configured to measure the supply voltage value present at its regulators, which can be read back to the microcontroller over the serial interface. The result is a 7 bit value can then be converted to voltage using Equation 5 [14].

$$Batt\ Voltage = \frac{Value}{21.1}$$

**Equation 5.** Battery Voltage Conversion from Read Back Value

The latter method was chosen for the pillbox as it only requires additional software to be added to the microcontroller eliminating any of the resistors needed for the voltage divider for the first method. Figure 27 shows the details of the battery monitoring function execution.



**Figure 27.** Battery Monitoring Functionality

### 3.4.9 RF Communication

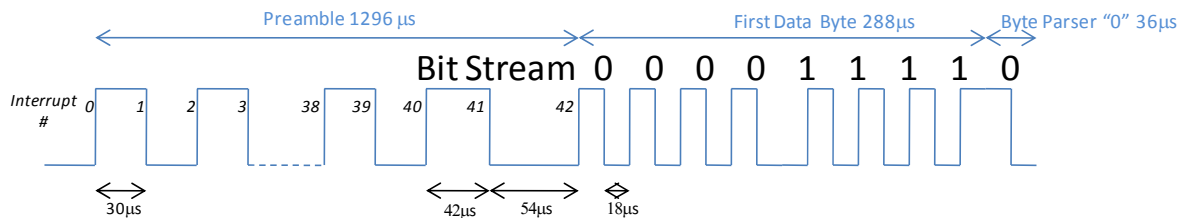
The RF Communication requires precise timing to provide the pulse widths needed to fulfill the ISO 18000 part 7 standards. The microcontroller is responsible for generating the stream of bits

with the proper timing and format. This section focuses on formation of the binary stream, its format, and the timing values needed.

### 3.4.9.1 Standards

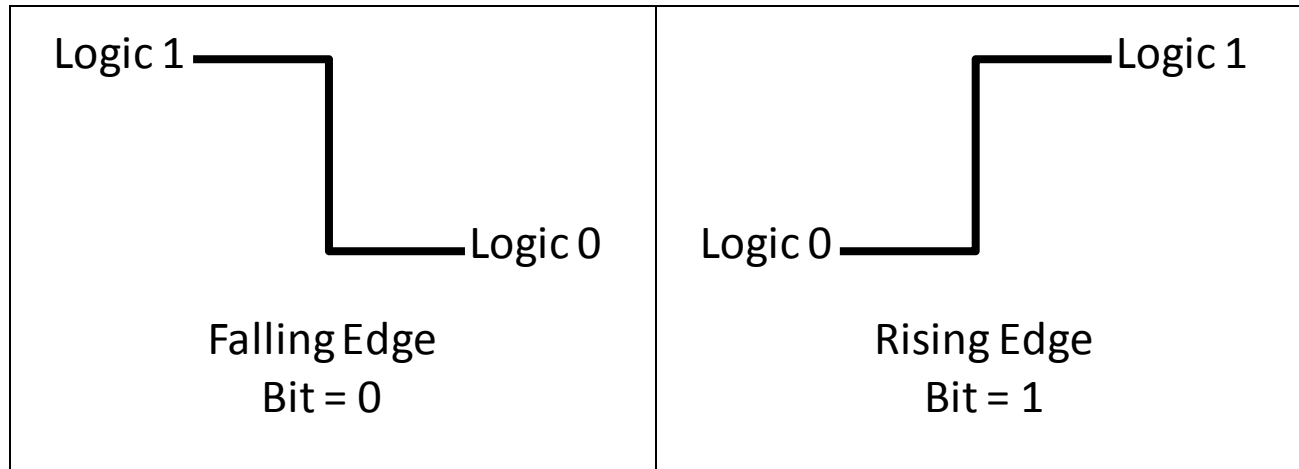
ISO 18000 part 7 specifies two types of devices: a receiver and a tag. The tag operates as a slave device and does not transmit unless asked to by the receiver. The pillbox is similar to a tag, but the receiving functionality has been removed. This is because the ADF7020 draws a considerable amount of current when transmitting or receiving. The transmitting current draw is limited however as it only sends one message and then powers down the ADF7020 and the microcontroller enters sleep mode. The receiving current draw is slightly lower, but the device must listen for periods up to 30 seconds requiring a considerable amount of power from the battery. In order to optimize the power consumption, the receiving functionality was removed.

The bit stream structure of a tag transmission is shown in Figure 28 below. The preamble precedes the actual data with a series of 30us pulses. The last pulse transition of the preamble is longer to signal the end of the preamble and that data bytes will be sent next. The data bytes require 18us interrupts to provide the appropriate bit rate as shown. The bytes are Manchester encoded with each byte separated by a “0” bit parser.



**Figure 28.** ISO 18000 part 7 Timing Diagram for Tag Transmission

Manchester encoding has different variations which classify the rising and falling edges of transitions as associated bit values. ISO 18000 part 7 specifies that a falling edge from logic 1 to logic 0 is a 0 bit and a rising edge from logic 0 to logic 1 is a 1 bit. Figure 29 shows an illustration of the Manchester standard being utilized.

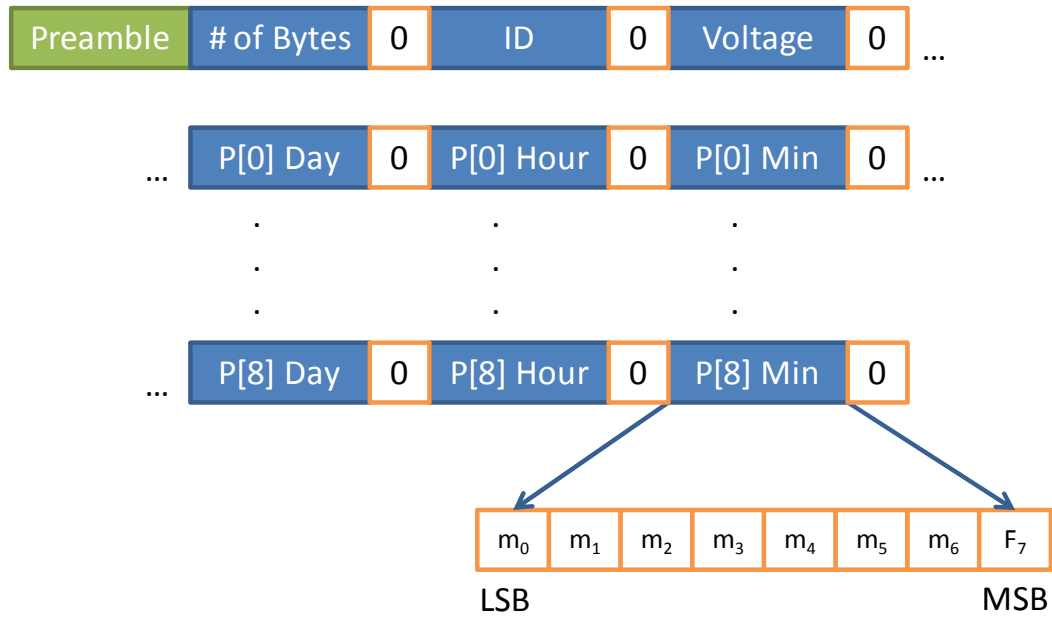


**Figure 29.** Manchester Encoding Standard

### 3.4.9.2 Packet Format

The data packet format for the pillbox matches the format of the ISO 18000 part 7 read memory command. The first byte of data in a transmission contains the number of bytes following it in the message. This provides a maximum message length of 255 bytes. Figure 30 shows the structure of the of the pillbox data packet with each byte field labeled. The preamble is considered a consolidated byte field because it is essentially a string of 20 Manchester 0's with a varying bit rate (interrupt intervals). The first three data bytes provide a header with the number of bytes in the message, ID number of the pillbox, and voltage level of the battery. Every three byte group after these header bytes correspond to a pill within the pillbox with each byte corresponding to a RTC status for that pill. If a pill has not been removed, all three fields will be

0. If a pill has been removed, the RTC value of the pill removal event is sent, and the MSB of the Min byte is set to 1 as a flag indicating that the pill has been removed. This flag bit was utilized for coding convenience in both the microcontroller and receiver and does not affect the RTC values as only 6 bits are required to represent the number of minutes that have passed. The packet and byte structure is shown in the Figure 30 below.



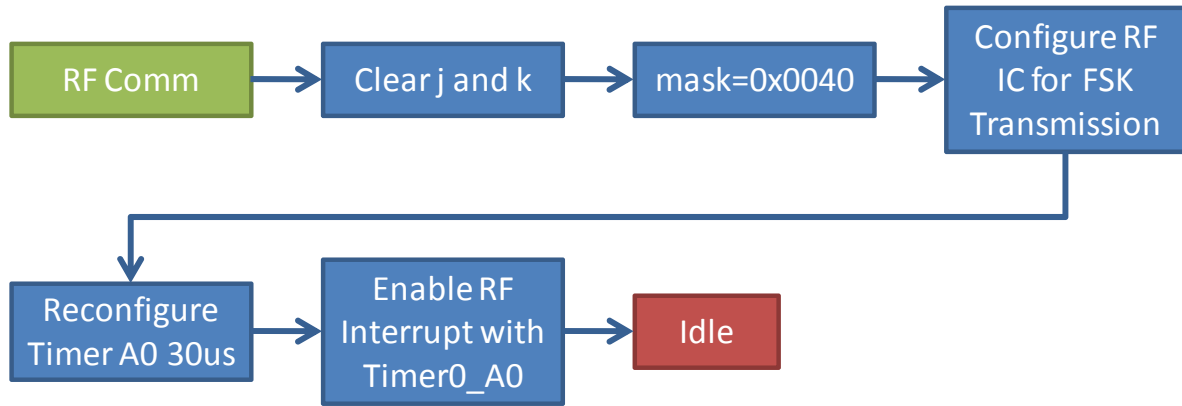
**Figure 30.** Transmission Packet Structure

### 3.4.9.3 Communication Program Flow

The RF Communication task is executed by first configuring the timer to provide 30us interrupts for the preamble pulse sequence. The bytes are stored in an array and are transmitted by bit banging the values of the array into the bit stream. This requires the use of index values to track how many transitions have been occurred for the preamble ( $j$ ), the current array byte being transmitted ( $k$ ), and which bit of the byte is currently being transmitted ( $mask$ ). Figure 31 shows the RF communication functional diagram illustrating the variable initializations and



configurations required for the RF interrupt timing. Once again, the process ends in an idle state as Timer0\_A0 interrupt provides the actual critical communication timing.



**Figure 31.** RF Communication Functional Diagram

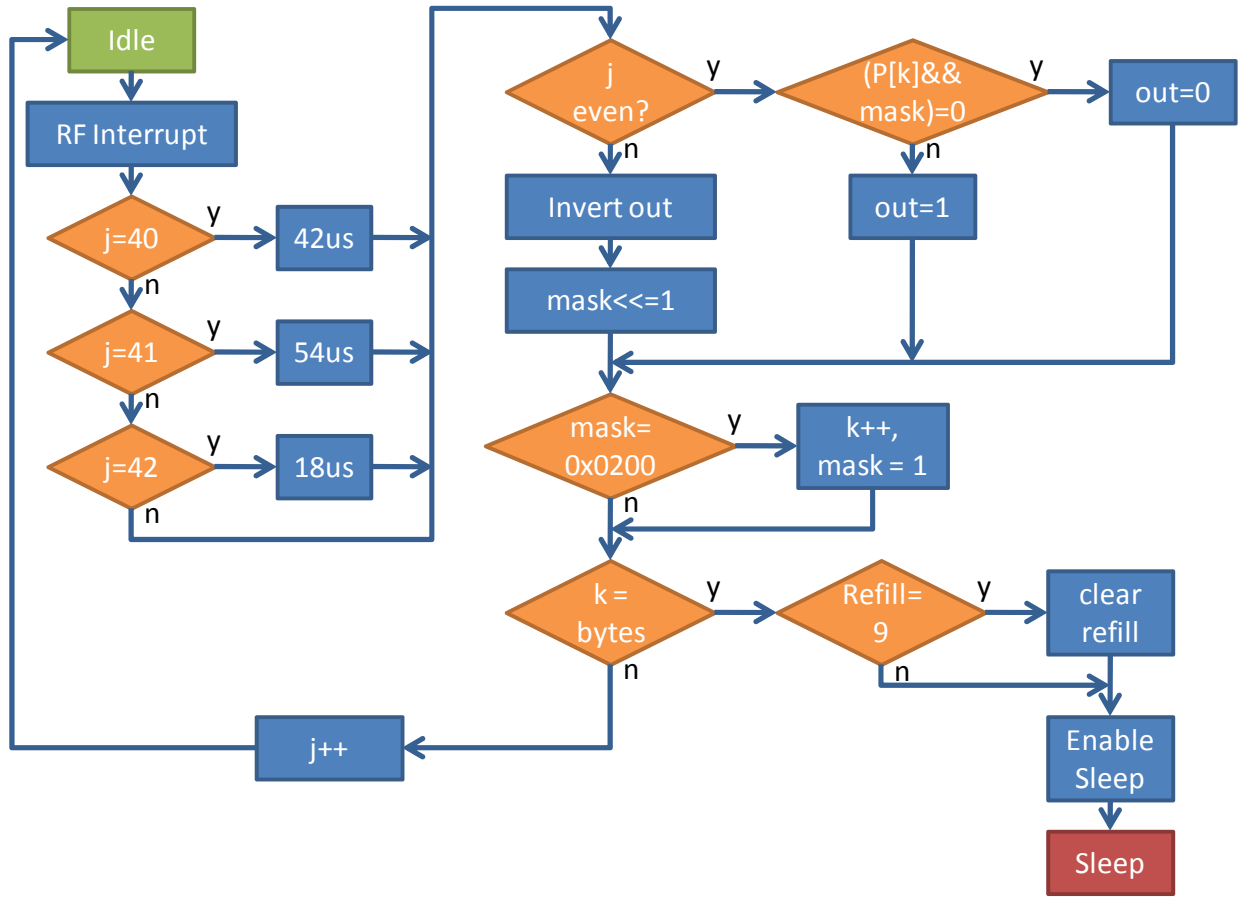
The details of the RF interrupt routine are shown in Figure 32 below. The process first checks to see how many RF interrupts have occurred by monitoring the  $j$  variable to determine when to change the bit rate to match the longer pulses at the end of the preamble and the data rate pulses as explained in Figure 28. The interrupt interval rate is changed on the fly allowing for a smooth pulse width transition.

ISO 18000 part 7 specifies that a Frequency Shift Keying (FSK) logical 1 is the negative frequency deviation from the carrier which is unusual and inverted from typical FSK implementation. As such, the output of the microcontroller must be inverted to provide the proper logical slope transitions. This is because a digital 1 from the microcontroller will generate a positive frequency deviation or a logical 0 according to the standard.

The Manchester encoding is performed by first determining if the interrupt is occurring in the first half of the Manchester bit transmission, or the latter half. If  $j$  is even, the first half of a Manchester bit must be sent, which requires the analysis of the current byte in the array to be masked with the *mask* variable. The mask isolates the correct bit to be transmitted from the byte

in the array. If the result of this masking operation is 0, the bit to be transmitted is 0 and the microcontroller should set the *out* variable to 0 providing the first half of a rising edge in frequency deviation (falling edge in logic). Otherwise, the bit to be transmitted is a 1 and the microcontroller should set the *out* variable to 1 providing the first half of a falling edge in frequency deviation (rising edge in logic).

If *j* is odd, the process becomes simple as the output simply needs to be inverted from the previous *out* value providing the appropriate slope for the Manchester encoding. The algorithm then bit shifts the *mask* variable to the left to provide the appropriate *mask* for the next bit to be transmitted. The *mask* variable resets after 9 bits are transmitted in order to allow one full byte to be transmitted plus an additional 0 bit that is required by the ISO 18000 part 7 as shown in Figure 28. Each time the *mask* variable resets, the *k* variable increments to index the next byte of the message to be transmitted. Once *k* matches the *bytes* variable, all bytes that should have been transmitted have been. If the *refill* condition had occurred, it is cleared and the *byte* variable is changed from the short packet length value (2) to the normal long packet length value (29). The sleep mode is then re-enabled and entered to save power until the next interrupt event.



**Figure 32.** RF Interrupt Functional Details

### 3.5 RF INTEGRATED CIRCUIT

The RF Communication was implemented by utilizing the ADF7020 IC. This chip is an RF transceiver capable of transmitting and receiving over various RF bands and can be configured to utilize a variety of modulations including FSK. The chip itself requires an extensive collection of passive components for regulators, filters and control loops along with an 11.592MHz crystal to provide the reference for carrier frequency generation. The chip itself is widely configurable

and utilizes a programming serial interface for a microcontroller to access its functional registers. These registers define the operational RF parameters used by the ADF7020. The datasheet describes each of these functional registers in detail and their use in various applications [14].

A separate data serial interface provides a connected microcontroller with the received data from the ADF7020 or an interface to provide the ADF7020 with desired data to be transmitted. The pillbox is currently implemented as a “transmit only” device, so the data interface is driven by the output of microcontroller. An enable pin controls the ADF7020’s power, and allows it to be turned off when not in use to save power, but the device’s functional registers must be reprogrammed when the chip is re-enabled.

### **3.5.1 Configuration**

The ADF7020 was configured to provide an FSK modulated signal with a 433.92MHz carrier and a frequency deviation of 50kHz. The proper ADF7020 functional register values for the desired configuration were challenging to determine when only using the datasheet due to the large number of options. ADI however provided a software tool to aid in this process, which was released with their evaluation boards and available free for download. The desired RF communication parameters were entered into the tool, and the needed functional register values for the ADF7020 were returned in hexadecimal format.

A C function was developed for the microcontroller that would manage the programming serial data connection for the ADF7020 by applying the appropriate enable and clock signals. The hexadecimal register values were then supplied to the function as long integer arguments, which could be copied directly from the software tool. The function decomposed the integer into

a bit stream using logical operators and bit masks with a simple loop and bit bangs the values over the serial interface.

Once configured, the data serial line can be driven by a microcontroller with a bit stream message. The bit rate of the message is determined solely by the bit rate of the data serial line leaving the timing up to the microcontroller. The ADF7020 interprets logic 1 as +50 kHz output frequency and logic 0 as -50 kHz signal.

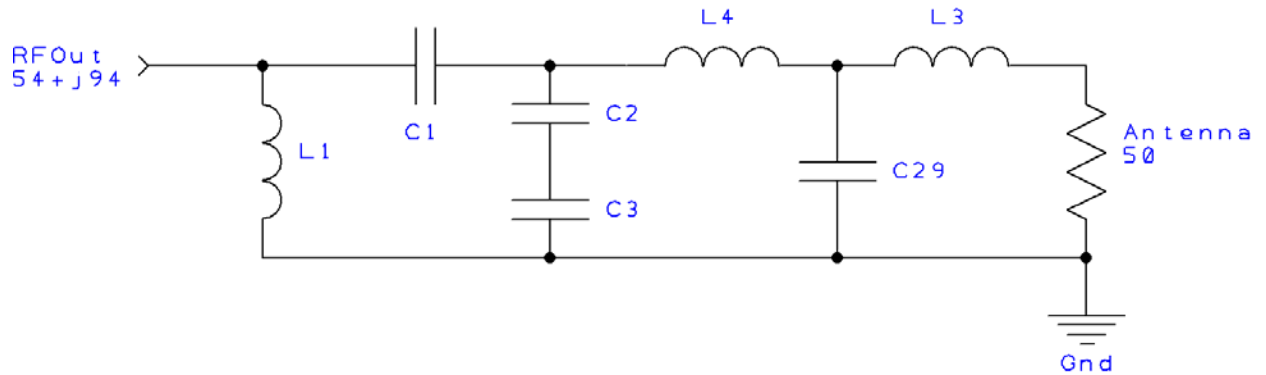
Monitoring the RF transmissions of the device with an RTSA verified that the device was providing RF signals within the desired frequency range, but the center carrier frequency was slightly offset. This could easily be explained by slight frequency errors within the reference crystal. In order to correct this offset, the resulting frequencies from a constant logical 0 (carrier + 50 kHz), and logical 1 (carrier - 50 kHz) were measured independently to provide the needed offset value. The value of the appropriate ADF7020 functional register was then modified to shift the carrier frequency to its desired value.

### **3.5.2 Antenna Matching Network**

The ADF7020 evaluation board documentation supplied information not only about the needed passive components for the chip, but also an optimized RF layout and matching network designed by ADI. Two problems arose when trying to implement this layout: the board thickness used by the design is not commonly available, and the required board space was not available for the pillbox PCB. As such, the design's component values were still used, but the layout had to be done by hand to fit the space constraints.

The major concern of the layout was the RF output to the board's matching network and antenna. The evaluation board's design allowed both transmit and receive functionality to be

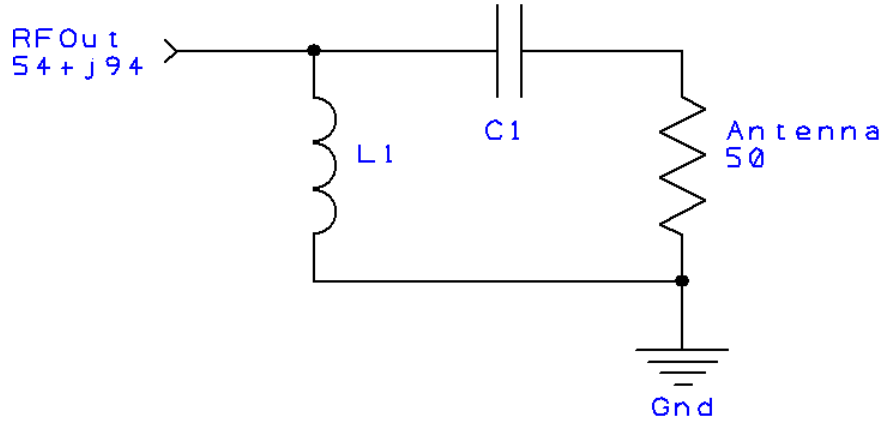
used with a single antenna and matching network. Unfortunately, the power amplifier for the transmit circuitry has a different output impedance than the low noise amplifier of the receive circuitry. As such, the designed matching network is a compromise between the two providing adequate performance for both. The equivalent matching network for the transmitter is shown in Figure 33 with component values and identifiers matching those given in Appendix C. The optimal input impedance to the network for the transmitter is  $54 + j94$ , but the calculated impedance of this RF network at 433.92MHz is  $6.4 + j44$ . The matching is mainly controlled by the first half of the network with L3, L4, and C29 providing a harmonic filter that passes the carrier frequency band to the antenna.



**Figure 33.** Reference RF Output Network for Transmitting

Due to a slight error in the layout of the RF matching network, a right angle corner existed in the microstrip connecting the ADF7020 to the antenna. This caused large RF transmission power losses for the board making communication impossible. This problem was minimized by moving the antenna's position slightly to the left of its original position on the board and connecting it to the micro strip where the right angle occurred. This modification required that both the harmonic filter and receive functionality be removed from the board in

order to allow two component pads to be available for a matching network. Figure 34 shows the design chosen for the matching network.



**Figure 34.** Designed RF Output Network for Transmitting

The values of L1 and C1 were determined by solving following system of equations based on the impedance calculations of the network. Equation 6 represents the real part of the impedance and Equation 7 represents the imaginary part of the impedance.

$$54 = 800 \left( \frac{\pi^4 f^4 L_1^2 C_1^2}{1 + 10000 \pi^2 f^2 C_1^2 - 8 \pi^2 f^2 L_1 C_1 + 16 \pi^4 f^4 L_1^2 C_1^2} \right)$$

**Equation 6.** Antenna Matching, Real Part of Impedance

$$94 = 2 \left( \frac{\pi f L_1 (1 + 10000 \pi^2 f^2 C_1^2 - 4 \pi^2 f^2 L_1 C_1)}{1 + 10000 \pi^2 f^2 C_1^2 - 8 \pi^2 f^2 L_1 C_1 + 16 \pi^4 f^4 L_1^2 C_1^2} \right)$$

**Equation 7.** Antenna Matching, Imaginary Part of Impedance

Maple was utilized to solve these coupled equations resulting with C1 being 4pF and L1 being 22nH, which are both available and reasonable values. The next design iteration of the board should correct the layout allowing the harmonic filter to be included in the matching network.

### **3.6 POWER CONSIDERATIONS**

The entire pillbox must be portable and powered by a battery capable of lasting at least a month. The choice of battery was difficult because the size of the battery and the amount of power available are two conflicting constraints. Given the tight case dimensions and the operational profile of the device, a small CR2032 lithium coin cell battery was chosen for the pillbox. These batteries provide 3V, but only at very limited current rating and are typically used in devices such as watches.

A data sheet for a CR2032 battery shows that the standard load current provided by the battery is only 0.2uA [23]. Luckily, the batteries can supply large pulse currents in the mA range for very short periods of time providing the needed power for the active state of the microcontroller and the RF transmission. The pulse current ability of these batteries and their lifetimes are shown in their use in wireless car remotes for locks and alarms.

All components chosen for the pillbox were selected so they could operate using a single 3V battery supply eliminating the need for a voltage regulator. This saves both board space and power losses from the regulator. The ADF7020 chip has the highest voltage requirements of all of the chips requiring 2.3V to operate. For this reason, the battery monitoring functionality triggers an indicator once the voltage drops below 2.5V.



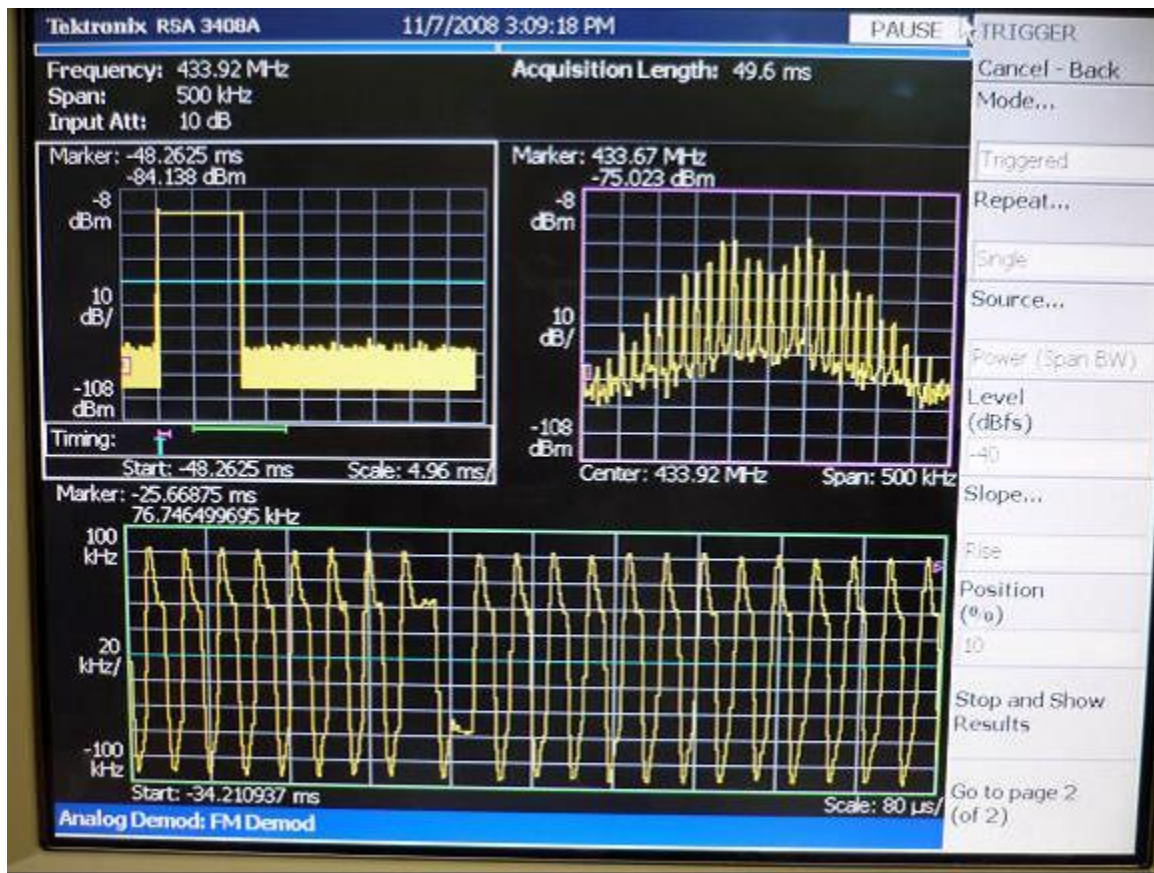
Initial testing of the pillbox when connected to a power supply provided insight into the total current consumption of the device. When the pillbox is configured for constant RF transmission, the power supply shows that approximately 45mA of current are used. This is higher than expected as the ADF7020 datasheet expects only 20mA to be drawn with an additional 10mA from other sources including the microcontroller. This may be due to the modified matching network and the high gain settings of the ADF7020.

### **3.7 RECEIVING STATION**

The pillbox was designed to operate under similar protocols as the RFID ISO 18000 part 7 specifications. Unfortunately, being a “transmit only” device makes it currently incompatible with commercial receivers. As such, a receiving station needed to be designed in order to provide a fully functional system for this proof of concept.

#### **3.7.1 Real Time Spectrum Analyzer**

The RTSA proved invaluable in the design and verification of the pillbox communication and RF standards. The Tektronix 3408A is a powerful tool capable of not only providing spectrograms, but also demodulation functionality. The demodulation options included both Frequency Modulation (FM) and FSK settings. An example of the FM demodulation of the pillbox transmission is shown in Figure 35 below.



**Figure 35.** RTSA FM Demodulation

There are three different windows on the RTSA display. The top left window shows the relative RF energy detected in the band vs. time. This provides an effective method of determining if an RF signal is being transmitted as a sharp rise in power can be observed as shown in the figure as a pulse. The trigger settings utilized to test the pillbox use the results from this window to determine when to begin sampling. The horizontal blue line at -48dBm shows the trigger threshold. The second window shows the spectral content of the captured signal, and the bottom window shows the demodulated waveform. The demodulated waveform provides an illustration of the FSK waveform showing both positive and negative deviations from the carrier frequency. FSK is simply a two level FM. In this figure, the preamble, and the beginning of

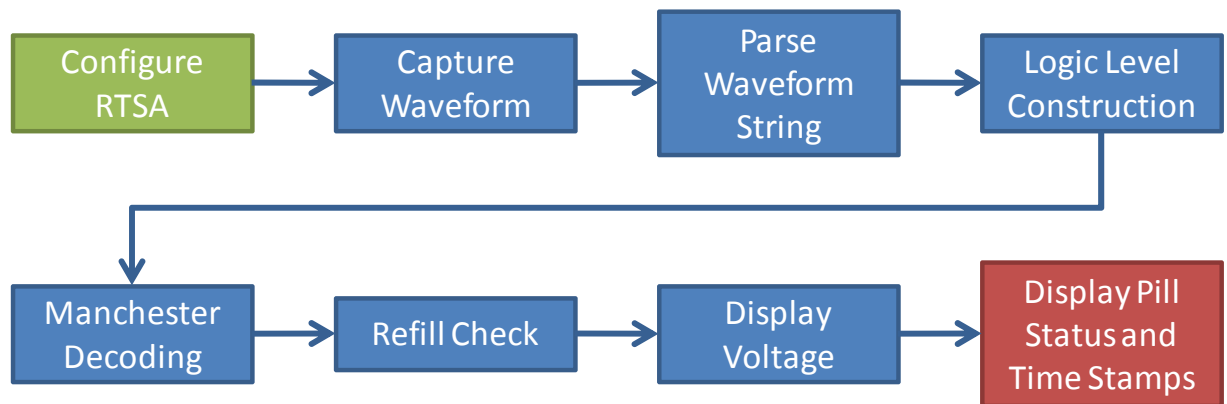
data transmission are shown in the waveform window, as the longer pulse widths at the end of the preamble are clearly visible in the first half of the screen.

The FSK demodulation display is similar to the FM with the waveform window being replaced with the symbol map showing the resulting bit stream of 1's and 0's. This setting requires more setup, as the parameters of the FSK modulation must be entered including bit rate and modulation type. This causes problems with the given transmission standard being used by the pillbox as the preamble is technically a string of 0's at a different bit rate from the data section leading the RTSA to display "x" values for unknown decoded bits. This makes determining when the data stream starts difficult and can lead to demodulation problems if the RTSA tries to decode the wrong edge of the data stream causing the resulting bit stream to be inverted. Another problem found with the FSK demodulator was that if the bit rate was not set to the exact bit rate being transmitted, the decoded data would become corrupted after so many bytes are transmitted. Testing showed that this would occur with the pillbox after transmitting more than 8 bytes of data. The current prototype requires that 30 bytes of data be sent reliably, which excludes the FSK demodulator functionality to be used.

In order to provide an HMI and to verify the operation of the pillbox transmission code, a computer based interface was needed to capture, demodulate, and decode the message sent. The RFID lab's work has been mainly focused on the conformance of equipment to standards such as ISO 18000 part 7, but the desired functionality for the pillbox receiver was not available as a current lab testing process. The ideal receiver for the pillbox would listen at all times, and capture and analyze a waveform when a trigger event occurs such as the overall RF power rising similar to the triggering functionality available with the RTSA. Because of the RTSA's ability to trigger an automatic sampling process, it was used as a key piece of the receiving station.

### 3.7.2 LabVIEW

LabVIEW is a graphical programming language and software package which is ideally used to create interfaces between existing hardware and lab equipment. This process is very efficient as National Instruments provides a large library of drivers and example code for an extensive collection of devices. The Tektronix TKRSA3408a RTSA driver was downloaded and its basic functionality was tested by using a GPIB interface between the computer and RTSA. A vi (LabVIEW function) was created to setup a triggered FM demodulation measurement and to capture the resulting waveform. This waveform was then processed and decoded in LabVIEW to store and illustrate the current state of the pillbox. Figure 36 shows the functional overview of the receiver.

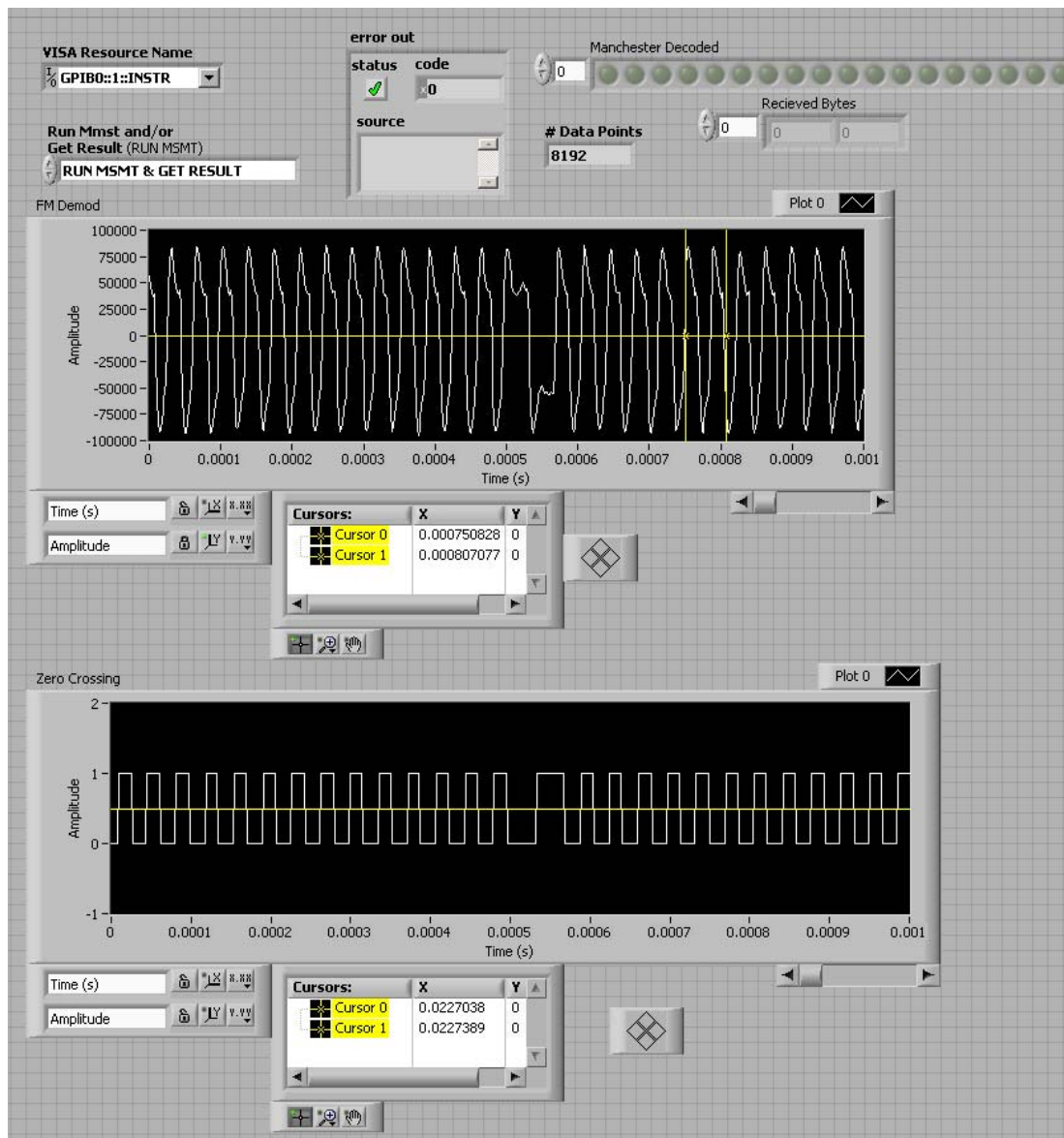


**Figure 36.** Receiver Functional Overview

#### 3.7.2.1 Logic Level Data Stream Recovery

Given the ISO 18000 part 7 standards and timing specifications, a procedure was developed in LabVIEW to properly identify, decode, and display the transmitted data from the pillbox. The sampling frequency of the waveform from the RTSA was measured to be 640kHz providing a

time scale for the captured waveform. The FM demodulated waveform is a graph that shows frequency deviation from the carrier over time. Positive values of the waveform correspond to logical 0 and negative correspond to logical 1. In order to reconstruct the logical data stream, a simple less than 0 conditional statement was used which translated the waveform to logic level 0's and 1's. Figure 37 shows the LabVIEW front panel display showing both the captured FM demodulated data on the top graph and the reconstructed logical data stream on the bottom.

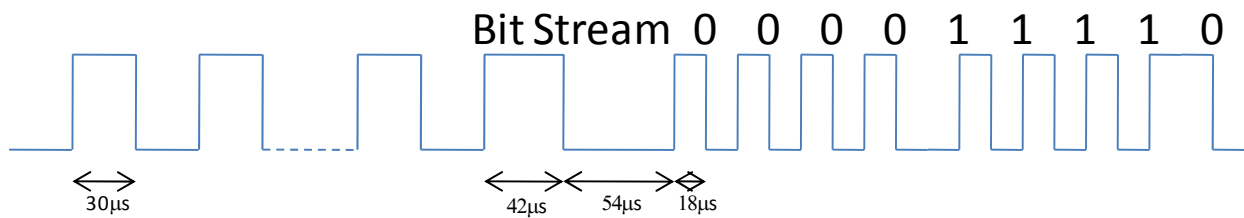


**Figure 37.** LabVIEW Interface to RTSA, Top Waveform Raw FM Demodulation, Bottom Bit Stream

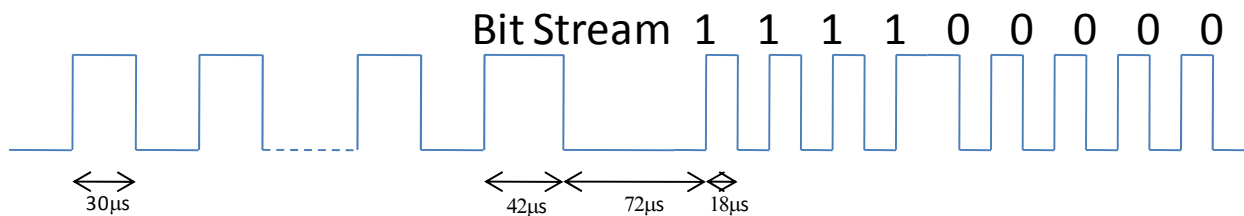
### 3.7.2.2 Manchester Decoding

The logical data stream is still Manchester encoded and must be transformed into a bit stream in order to be interpreted. This can be performed by analyzing the pulse widths of the logical data stream. The longest pulse of the data stream occurs at the end of the preamble with it being 54us. This pulse however may be longer if the first bit of data following the preamble is a 1 as it

requires a rising edge 18us later. By comparing the length of the pulse after 42us, the first data bit of the message can be determined. Figure 38 and Figure 39 show the differences between the logical data streams to show this pulse timing.



**Figure 38.** Example Waveform, First Bit Transmitted is a 0

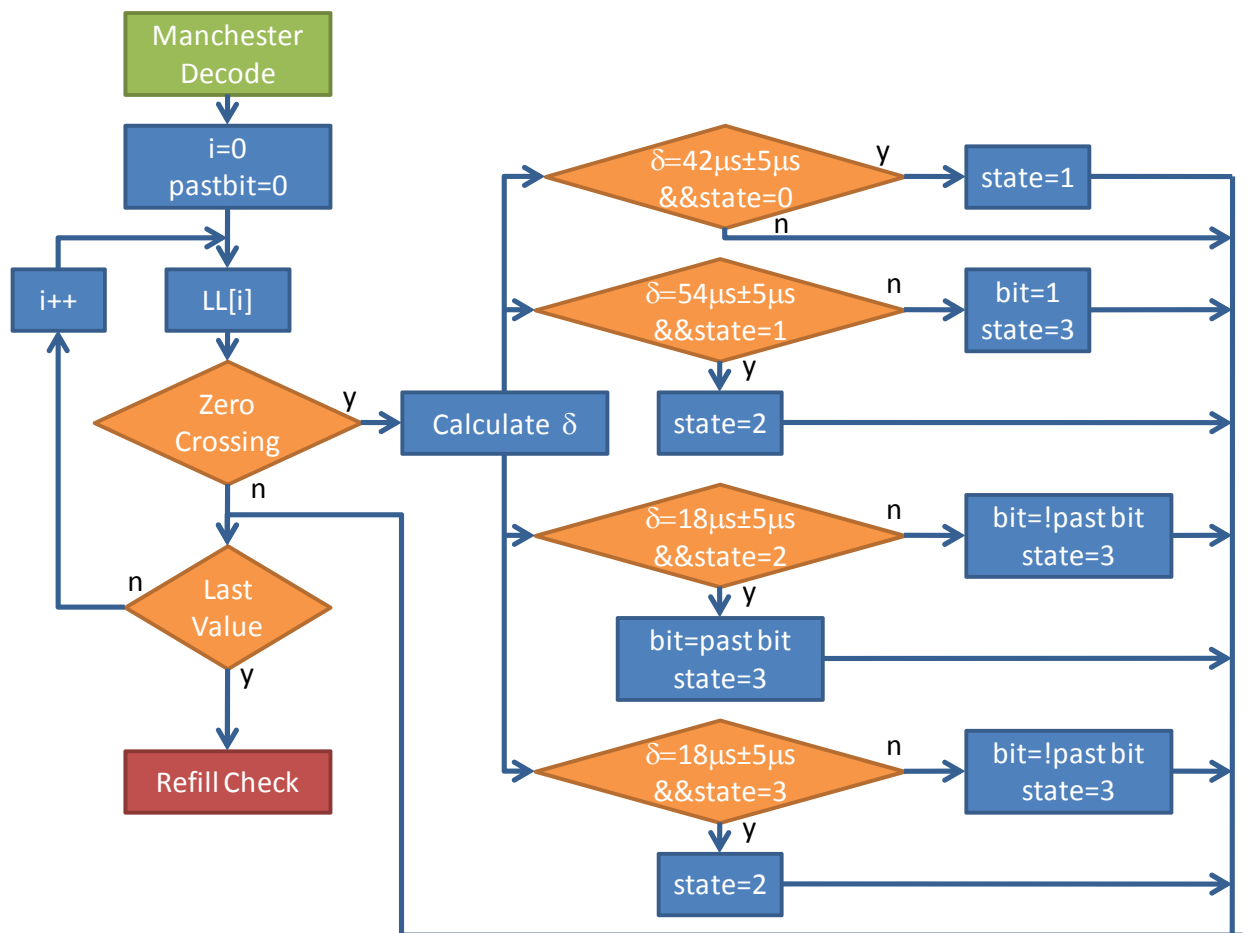


**Figure 39.** Example Waveform, First Bit Transmitted is a 1

The timing pattern of the Manchester encoded data message can be found by looking at the previous figures. If the time between pulses remains short at 18us, the resulting bit every 2 transitions will be the same as the previous bit value. When a longer pulse is measured at 36us, the output will be the opposite of the previous bit value. This simple algorithm was implemented with a buffer to accept pulse values within 5us of the desired pulse width to permit inaccuracies with the transmission bit rate and sampling frequency resolution effects.

Figure 40 shows the organization of the Manchester decoder where  $\delta$  is the measured pulse width and  $LL$  is the array storing the Logic Level data stream. The algorithm has four states. *State 0* is what the algorithm is initialized at, and it checks to see if the 42us pulse

requirement is met from the preamble. Once the condition is met, *State 1* is entered which checks to see if the next pulse is 54 $\mu$ s. If it is, the *state* is incremented to *State 2*, otherwise a rising edge from a 1 bit has occurred, and *State 3* is entered. Once in either *State 2* or *State 3*, the algorithm will jump between the two to provide the decoded message. *State 2* can be thought of as an active state in which no matter what the measured pulse width, a bit will be decoded. *State 3* can be thought of as a waiting state in which only the long pulse will cause a bit to be decoded.



**Figure 40.** Manchester Decoding Algorithm Using Pulse Widths



### 3.7.2.3 Data Presentation

Once the bit stream has been reconstructed, the packet can be translated using the packet format illustrated in Figure 30. The first byte of data defines how many bytes of data follow in the packet. If only 2 bytes follow, the receiver knows that a refill condition has occurred. The receiver then saves the current system time to an excel file on the computer and lights up a status indicator to show that the time reference has been updated. This is a simple implementation and future designs will likely use a database to store such information.

If 29 bytes were sent, a normal pillbox status message has been transmitted providing information about which pills have been removed and when. The receiver HMI has LED indicators arranged to represent each pill in the pillbox, which remain lit when the pill is still in the case. Once the pill has been removed, the associated LED turns off and the time stamp of the removal is shown below the LED. The timestamp is calculated by adding the RTC values from the pillbox to the refill timestamp stored on the computer. The final indicator on the receiver is the battery status measurement. This measurement shows the current voltage level of the battery, and a warning indicator if the voltage drops below 2.5V.

Figure 41 shows the entire LabVIEW interface developed for the receiver. The left half of the interface was used extensively in the testing phase of the project verifying pulse widths and the accuracy of the Manchester decoding algorithm. The right half of the display is ideally what the final system HMI would resemble quickly showing the current status of the pillbox.



**Figure 41.** Entire LabVIEW Interface for Receiver

## 4.0 PROTOTYPE ASSEMBLY

The prototype assembly required careful attention to details and dimensions to ensure that the project would come together and operate successfully. This was especially true for the alignment of the optical components with the pill holder and their placement within the pillbox. The Solidworks model provided invaluable information needed to size the PCBs and the placement of components on the layout before any case modifications were needed. Figure 42 shows a disassembled view of the resulting prototype pillbox.



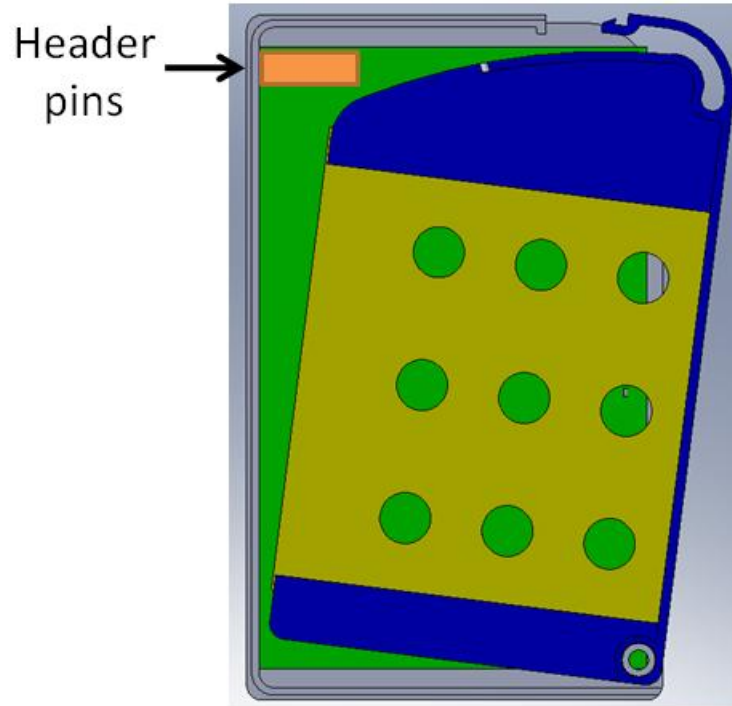
**Figure 42.** Disassembled Prototype Pillbox

## 4.1 BOARD LAYOUT

The PCB board dimension for the top and bottom was determined to be 2.5 inches by 4 inches. This was carefully decided as the square edges and corners of the board needed to fit flush with the interior case edges, which has rounded corners. Since two smaller boards were needed, only one layout pattern was completed using a larger 6 inch by 4 inch standard size board with both the top and bottom layouts distributed on opposite ends. The boards could then be easily cut apart using a band saw in the prototyping shop.

A schematic of the circuit layout was generated before any layouts were attempted. This provided help with the layout process as PCB Artist (the layout design tool) provided indicators showing which pins should be connected. Both the top and bottom board schematics were drawn onto one schematic, but each with their own power and ground busses. This avoided the layout software generating errors when trying to connect the power and ground busses with a trace on the board instead of through connectors. The schematic also allowed more detailed design rule and error checks to ensure that the entire schematic netlist is satisfied by the layout.

One of the most important design challenges of the pillbox was to determine how to electrically connect the top and bottom PCB's for both power and communication. The center pill holder has a slight curved taper along the top inside edge to allow the center piece to swing open and clear the top edge of the pill case. This taper leaves a small area inside of the case which is not covered by the pill holder when closed providing a space for an interconnect. A small 10 position, 2 row header pin connector can be placed in this space allowing the two boards to simply slide together. The Solidworks model once again provided the ability to ensure that the size of the header pins and their placement wouldn't obstruct the motion of the pill holder by monitoring the path of the pill holder as shown in Figure 43



**Figure 43.** Placement of Board Interconnect Header Pins

This header pin interconnect also serves a secondary purpose for the device. The MSP430 programmer utilizes a similar header pin interface with 14 positions as described in the programmer's documentation [12]. The documentation provided schematics and instructions on how to layout the programming port to use the minimum amount of pins by using their Spy-Bi-Wire JTAG connection. This connection requires 6 pins for programming including power and ground connections leaving the other 4 pins available for interboard communication. The MSP430 software can now be programmed by simply pulling the boards apart, and sliding the programmer over the header pins on the top side (microcontroller side) of the board. Figure 44 illustrates how the programmer is connected to the board.



**Figure 44.** MSP430 Programming Connection

The layout was done by hand with the optical components precisely placed to match the pill holder holes designed in the Solidworks model. The majority of the components were placed within an inch of the board edge along the sealed long side of the case. This was done for multiple reasons, but primarily to keep the optical array area of the boards free of any non optical components. The goal was to allow for future expansion of the optical area to accommodate denser, higher pill count blister packs. The components being placed closer to the board interconnect also made the layout simpler and the tracks shorter which is good design practice for RF boards.

Because the boards are transmitting RF, design rules were followed to ensure that the board itself would not act as an antenna. These rules included covering the board with a

grounded poured copper area on both the top and bottom of both boards except in areas near the antenna. This was to aid in shielding the parts of the board which were not RF related, while still providing an area for the antenna to propagate. Grounding vias were placed in a grid pattern of approximately 0.5 inches to connect the two sides of poured copper to satisfy the general  $\lambda/20$  rule to avoid the copper areas becoming unwanted antennas and absorbing the RF transmissions.  $\lambda$  is the wavelength of the generated RF and is 56.94 inches for a 433.92MHz signal. This leads to a maximum distance of 1.36 inches between grounding vias.

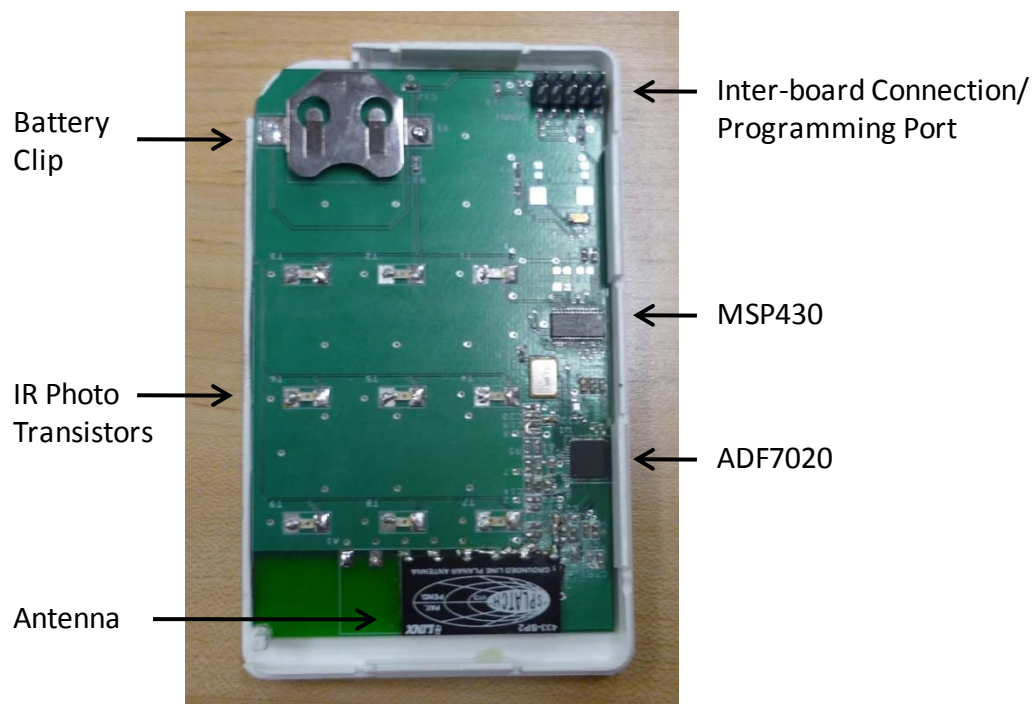
The pillbox was originally designed to utilize thinner boards on the order of 10 or 20 mils to allow the case to snap shut when assembled. This coupled with the fact that the boards needed to be mounted to the inside of the pill case led to the requirement for SMT type components only. This proved to be challenging for some components such as the header pins as they are typically through hole connections to provide more mechanical strength in the connection. Appendix C includes all of the information used to fabricate and assemble the electronics and boards for the pillbox including the schematic and associated PCB layout.

## **4.2 ASSEMBLY**

Once the board was received, it was first cut apart to separate the top and bottom PCB's. The boss connection location between top and bottom case were drilled out of the boards to allow them to slide into place. This was done precisely to allow the center pill holder to slide onto the boss and be held into place by the PCB's not the inside surface of the pill casing. The needed hole diameters and locations were once again determined from the Solidworks model and fabricated using a precise milling machine. This effort was required in order to allow the center

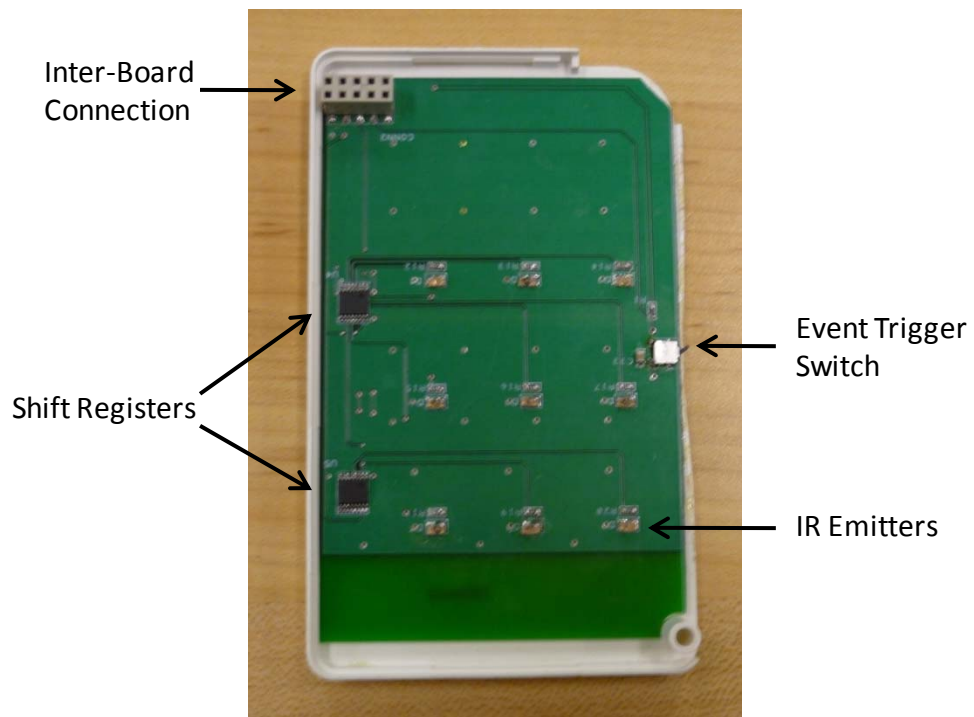
pill holder piece to easily slide in and out of the pillbox with enough clearance for the pills and the electronics.

The boards were then populated with components mainly by soldering them by hand following the schematics and layout given in Appendix C. The exception to this method was the ADF7020 chip, which is a QFN package that includes a grounding pad on the bottom center of the chip. In order to connect this chip, the reflow machine was required along with a great deal of patience to properly prepare and align the board. Figure 45 and Figure 46 below show the completed prototype PCBs mounted into their appropriate pillbox pieces.



**Figure 45.** Top Half of Pillbox with Populated PCB





**Figure 46.** Bottom Half of Pillbox with Populated PCB

## **5.0 FUNCTIONAL TESTING**

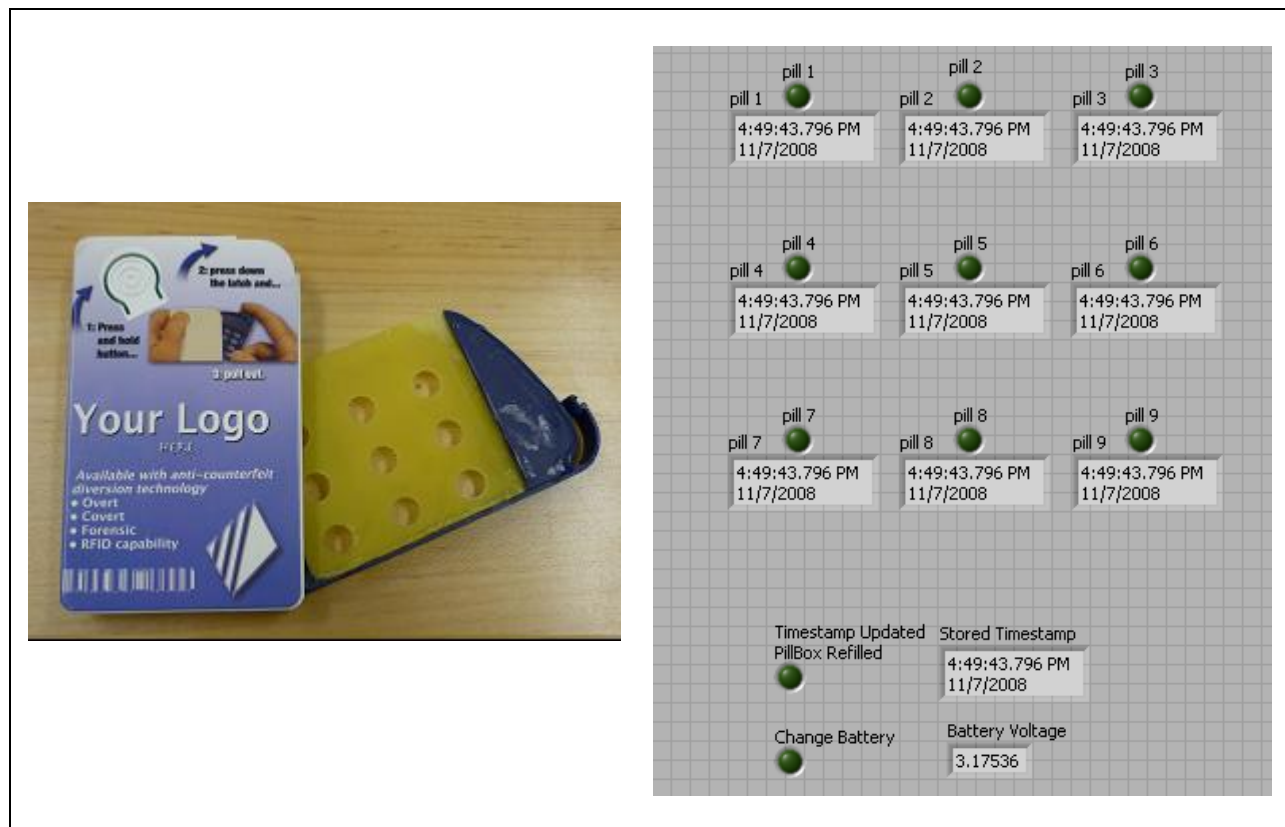
Testing the medication tracking system required both the use of the pillbox prototype and the developed receiving station. The behavior of the system was explored to demonstrate its operation and validate its functionality. Each of the subsections provides a different condition the pillbox can encounter and presents the results of the receiver's HMI. Not all combinations of pillbox conditions are presented as the general functionality is demonstrated by the chosen examples

### **5.1 EMPTY PILLBOX**

Figure 47 below shows the result from operating the pillbox with no blister pack or an empty blister pack. The receiving station displays that none of the pills are detected because all of the pill indicators are dim and show a timestamp. These timestamps are all the same since this measurement is taken as soon as the pillbox was first powered up with a new battery leaving the RTC values all as 0.

Below the pill indicators, the currently stored time stamp of the last refill condition is shown. This value only updates once a refill condition has been detected. If a refill condition is detected, the indicator next to the stored timestamp will light alerting the user that the receiving station has successfully detected the refill event and updated its records.

The final indicator at the bottom provides the measured battery voltage of the device. If the voltage drops below 2.5V, the Change Battery indicator turns red warning that the battery should be replaced.



**Figure 47.** Function Testing, Empty Pillbox

## 5.2 REFILL CONDITION

By placing a full blister pack into a previously totally empty device and closing it, the pillbox is able to detect that a refill event has occurred. Figure 48 shows the resulting display from the receiver for such an event. Notice that the pill indicators are all lit, and that the timestamps

below these indicators are hidden. This occurs because there are no valid RTC values to be displayed for any of the pills until a pill is removed.

The PillBox Refilled indicator is also lit indicating that the receiver correctly detected the refill event and was able to update the currently stored timestamp for the device. The resulting update can be seen by comparing the stored timestamp value in Figure 47 to the value in Figure 48. The value has updated to be 10 minutes later in the day.



**Figure 48.** Function Testing, Refill First Transmission

### 5.3 FULL PILLBOX

The full pillbox condition occurs whenever the pillbox is opened and closed after a refill, and no pills are removed. The pillbox detects that all of the pills are still present, and that a refill event has not occurred because an all empty condition did not precede the current detection. Figure 49 shows the receiver display for this situation with that all pill indicators being lit, but the refill indicator is not. This functionality prevents the pillbox from accidentally triggering a refill condition and resetting the RTC values when not at the pharmacy being refilled. This ensures the accuracy of the offset RTC values stored in the pillbox.



**Figure 49.** Functional Testing, Full Pillbox

## 5.4 ONE PILL REMOVED

A single pill is removed from the case, and the case is closed for the next test. The receiving station indicator clearly dims the pill that has been removed and provides a timestamp of when the event occurred as shown in Figure 50 below. The time stamp is the same as the refill time as this took place within the first minute of the refill procedure.



**Figure 50.** Functional Testing, 1st Pill Removed

## 5.5 TWO PILLS REMOVED

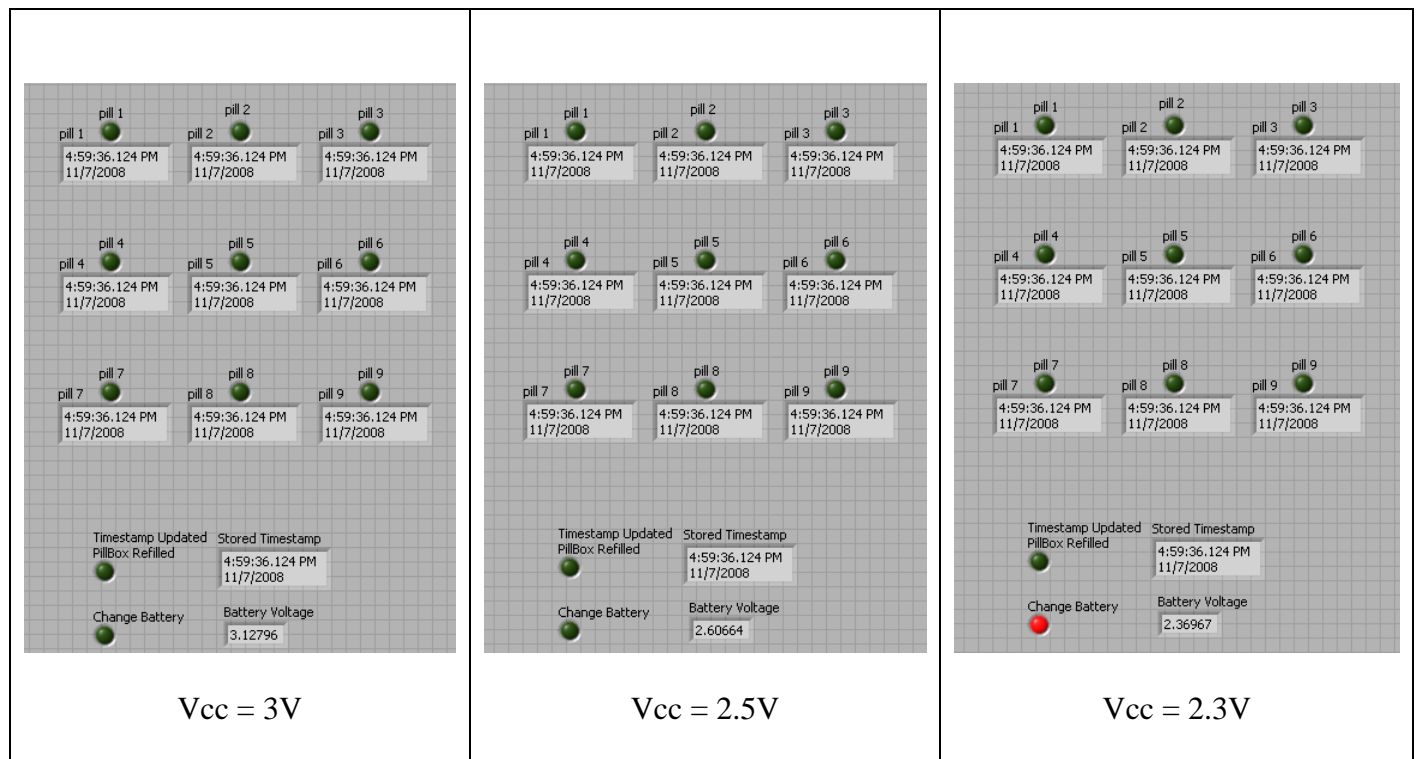
The next test was to remove a second pill one minute later and to observe the receiver's results. Figure 51 below shows the receiver display, which accurately detects the second pill location and a time stamp one minute later. This test demonstrates both the accuracy of the pill detection method and also the functionality of the timestamp system.



**Figure 51.** Functional Testing, 2<sup>nd</sup> Pill Removed

## 5.6 VOLTAGE MEASUREMENT TEST

In order to test the voltage measurement capability of the device, an adjustable power supply was used to power the pillbox via clip connectors instead of the coin cell. Three voltage levels were provided to the empty pillbox for the test: 3V, 2.5V, and 2.3V. The resulting receiver displays are shown in Figure 52 below. While the measured voltage measurement is slightly high compared to actual value, it does successfully detect the changes in the voltage. The Change Battery indicator does not trigger until a voltage value less than 2.5 was supplied as was designed showing that the battery monitor does operate as expected.



**Figure 52.** Functional Testing, Battery Monitor Validation



## 5.7 FUNCTIONAL CONCERNS

The designed system does function and provides a proof of concept pill detection technique, but small concerns were raised during the testing process. First, it was noticed that the device could easily transmit over 30 times meeting the requested test for battery life. Unfortunately, the device would stop transmitting and would not operate until a new battery was installed after approximately 50 transmissions. This would typically indicate a dying battery, however, the voltage level of the battery was still measured at 2.8V indicating that the battery still has sufficient charge. This nullifies the battery warning limit currently set on the device. The cause of the pillbox failure for these medium voltage batteries should be explored and a better warning system devised to provide early warning of failure.

A second concern is the RF transmission efficiency. The ADF7020 chip is configured for maximum gain to provide the 13 dBm of output power. Measurements with a power meter connected directly to the microstrip only result in -7dBm power source. This is disconcerting as over 15dBm of power is being lost, which may be due to the layout of the board itself. While currently functional, the devices effective transmission distance could be improved if this issue is studied further to determine the optimal RF layout for the given board size constraints.

## **6.0 CONCLUSIONS**

A proof of concept device has been developed and demonstrated that can accurately track individual pill removal from a compact and portable case that appears to be a normal pillbox. The device successfully uses common blister pack pill packages increasing the likelihood of adoption due to ease of handling by the pharmacy and ease of use by the patients. The pill detection method has been validated and has successfully been implemented in a compact and feasible form factor that can easily be expanded and configured to other blister pack dimensions. Wireless communication allows the device to communicate pill removal automatically eliminating any extra steps the user may need to take to operate the device other than removing the pills providing a simple and effective medication tracking solution.

## **7.0 FUTURE WORK**

The device's functional concerns have already been addressed with possible solutions suggested. The original board design has built in redundancies with pads for extra crystals and oscillators to be added in case the prototype's original design would not work. These pads were not required as the initial design worked and can be removed from the layout providing more space and flexibility with component locations aiding in the layout modifications for the next design revision. This is especially helpful as the ADF7020 component (or any RF IC) requires a great deal of board space given the number of passive support components needed for the chip.

The ADF7020 itself caused many problems with consistency during testing and was always the problem source if the device stopped transmitting. This was due to the solder connections coming loose with the QFN packaging and was fixed by simply reflowing the solder around the chip with the surface mount machine. In order to improve the reliability and durability of the device, a different RF IC should be selected with a package using leads to provide more surface area for the solder.

The pillbox is currently designed to operate with only 9 pills in a blister pack. A 28 pill blister pack should be obtained for a design reference, and the board modified in order to support the new package. This would likely require a larger pill case than the current pillbox provides. Given the Solidworks model has already been developed based off of the current prototype, a modified slightly larger version could be easily designed and built including a new single piece

pill holder. This would provide the opportunity to create a pillbox that can snap closed with the increased thickness introduced by the PCB's. The next prototype should also be built using the thinner PCB's available to reduce the overall weight of the device.

Studies should also be performed to analyze possible errors in the device including partially blocked empty pill locations from the back film holders of the blister packs. If this becomes an issue with the current design, different IR Emitters may need to be chosen to provide a broader beam angle to maximize the intensity over the desired pill aperture. This would need to be carefully balanced with creating a system that could cause false positive readings as discussed in section 3.3.2.

The receiving station should be redesigned to utilize the RFID readers currently being used in industry. This would eliminate the need for the RTSA to be used with the current receiver and ensure that the device could be used with available off the shelf reading devices. The receiver software would also need to be expanded to utilize databases to track and store the pill status data for different pillboxes replacing the simple timestamp storage method used by the current system.

This medication tracking system has many areas in which it can be generalized now that the basic operation has been shown to be feasible. Hopefully, this solution will become an attractive technology for the current medication adherence market and be will adopted for common use by the medical community.

## APPENDIX A

### MICROCONTROLLER SOURCE CODE

```
#include "msp430x21x2.h"

//Defined Bit Masks Used for ADF7020
#define P1_RFCE 0x01
#define P1_RFDATA 0x02
#define P1_RFLOCK 0x04
#define P1_RFDATACLK 0x08
#define P1_RFSCLK 0x10
#define P1_RFSREAD 0x20
#define P1_RFSDATA 0x40
#define P1_RFSLE 0x80
#define P2_BATT 0x01
#define P2_SWITCH 0x04
#define P2_SHIFTD 0x08
#define P2_SHIFTCLK 0x10
#define P3_SHIFTOE 0x01
#define P3_SENSOR 0x02
#define P3_RFMUX 0x04

void ADF7020(unsigned long values); //Configures ADF7020 Register Values
void BattCheck(void); //Reads Battery voltage from ADF7020
void TxClk(void); //Configures Timer Interrupt for precise RF transmission of 18000 part 7 Read Memory Response
void ShiftReg(void); //Uses shift registers to turn one IR Emitter at a time to detect pill status in package
void clockadjust(void);
void rtc(void);
//still need to add battery measurement

unsigned short clkcapdiff=0, clkcapnew=0, clkcapold=0, clkindex=0, clktarget=1464;
char day=0, hour=0, min=0;
unsigned char pillstaken=0, refill=0;
unsigned int i=0, j=0; //General index variables
unsigned int pillind=0; //index used for pill array
unsigned int bitmask=1; //mask used to bitbang serial communication with LSB first
unsigned int premcount = 0; //loop variable to track how many bits have been transmitted
unsigned char pill[90]={0,0,0, //First 3 bytes are for the preamble sequence
29,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0, //# of data bytes, Pillbox ID, Pillbox ID and Battery Status
//1,2,3,4,5,6,7,8,9,10, //2 bytes per pill location, 1st=day, 2nd=hour/min
//11,12,13,14,15,16,17,18,19,20,
//21,22,23,24,25,26,27,28,29,30,
31,32,33,34,35,36,37,38,39,40,
41,42,43,44,45,46,47,48,49,50,
51,52,53,54,55,56,57,58,59,60,
61,62,63,64,65,66,67,68,69,70,
```

```

71,72,73,74,75,76,77,78,79,80,
81,82,83,84};

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD;                // Stop watchdog timer

    //Power up Clock Configuration
    DCOCTL = 0x80;    //DCO=4, MOD=0
    BCSCCTL1 = DIVA_3 + 0x0C;    //LFXT1=LFM, ACLK/8, RSEL=12
    BCSCCTL2 = 0x00;    //MCLK=DCO=SMCLK, MCLK/1, SMCLK/1, Internal DCO res
    BCSCCTL3 = 0x08;    //LFXT=32768Hz, 10pF

    //Outputs SMCLK through 2.1 (pin 9) and ACLK through 2.0 (pin 8)
    P1DIR = P1_RFSLE + P1_RFSDATA + P1_RFSCCLK + P1_RFCE + P1_RFDATA;    //Sets Port1 GIO
    P2DIR = P2_SHIFTD + P2_SHIFTCLK;
    P3DIR = P3_SHIFTOE;

    //RTC initialization
    rtc();

    //Enable Interrupts for switch (Normal Operation)
    P2IE |= P2_SWITCH;
    P2IES &= ~P2_SWITCH;
    P2IFG &= ~P2_SWITCH;

    //Enable Interrupts for Powerup (Debugging)
    // P2IE |= 0x20;
    // P2IFG |= 0x20;

    _BIS_SR(LPM3_bits + GIE);    //enable global interrupts and low power mode, disable MCLK;

}//////////end main
void BattCheck(void)
{
    unsigned char batt=0;
    unsigned long mask=0x80000000;
    unsigned long values=0x00000108;    //Set ADC to on
    unsigned short loopind;
    for (loopind=0;loopind<32;loopind++)
    {
        P1OUT &= ~P1_RFSCCLK;    //Set SCLK Low
        if ((values & mask) == 0)
            P1OUT &= ~P1_RFSDATA;    //Set data bit 1 to low
        else
            P1OUT |= P1_RFSDATA;    //Set data bit 1 to high
        P1OUT |= P1_RFSCCLK;    //Set SCLK high
        mask>>=1;
    }
    //toggle SLE
    P1OUT |= P1_RFSLE;    //drive SLE high
    P1OUT &= ~P1_RFSLE;    //drive SLE low

    mask=0x80000000;
    values=0x00000157;    //Enable Battery Readback
    for (loopind=0;loopind<32;loopind++)
    {
        P1OUT &= ~P1_RFSCCLK;    //Set SCLK Low
        if ((values & mask) == 0)
            P1OUT &= ~P1_RFSDATA;    //Set data bit 1 to low
        else
            P1OUT |= P1_RFSDATA;    //Set data bit 1 to high
        P1OUT |= P1_RFSCCLK;    //Set SCLK high
        mask>>=1;
    }
    P1OUT |= P1_RFSLE;    //keep SLE high
    P1OUT &= ~P1_RFSCCLK;
    for (loopind=0;loopind<17;loopind++)
    {
        P1OUT |= P1_RFSCCLK;    //Set SCLK High

```

```

        batt<<=1; //Shift battery result
        if (P1IN & P1_RFSREAD) //if incoming bit is 1
            batt++; //add 1 to the resulting mask
        P1OUT &= ~P1_RFSCCLK; //Set SCLK high
    }
    P1OUT |= P1_RFSCCLK; //ignore last bit
    P1OUT &= ~P1_RFSCCLK;
    P1OUT &= ~P1_RFSLE; //drive SLE low
    pill[5]=batt;
}

void ADF7020(unsigned long values)
{
    unsigned long mask=0x80000000; //bitmask for ADF7020, moved MSB first
    unsigned short loopind;
    for (loopind=0;loopind<32;loopind++)
    {
        P1OUT &= ~P1_RFSCCLK; //Set SCLK Low
        if ((values & mask) == 0)
            P1OUT &= ~P1_RFSDATA; //Set data bit 1 to low
        else
            P1OUT |= P1_RFSDATA; //Set data bit 1 to high
        P1OUT |= P1_RFSCCLK; //Set SCLK high
        mask>>=1;
    }
    //toggle SLE
    P1OUT |= P1_RFSLE; //drive SLE high
    P1OUT &= ~P1_RFSLE; //drive SLE low
}

void rtc(void)
{
    TA1CCR0=61339; //60s = 61440*(4*(8/ACLK))
    TA1CCTL0=CCIE; //Enable Timer A1 interrupts
    TA1CTL = TASSEL_1 + ID_2 + MC_1; //upmode
}

void clockadjust(void)
{
    TA0CTL=TACLR;
    clkindex=0;
    clkcappdiff=0;
    clkcappnew=0;
    clkcappold=0;
    TA0CCTL2 = CM_2 + CCIS_1 + SCS + CAP + CCIE; //ACLK/8 trigger, rising edge
    TA0CTL = TASSEL_2 + MC_2; // SMCLK, continuous
}

void TxClk(void)
{
    TA0CTL=TACLR;
    TA0CCR0=180; //30us preamble pulses
    TA0CTL=TASSEL_2+MC_1; //MCLK source, up counter
    TA0CCTL0=CCIE; //Enable Timer A0 Interrupts
    bitmask=0x40; //Set bitmask so that 21 0's are sent for the preamble (2 bytes = 18 bits, 1 byte = 8 data bits + 1 0 bit)
    premcount=0; //Reset premcount
    pillind=0; //Reset pillind
}

void ShiftReg(void)
{
    unsigned char indoffset=0;
    refill=0;
    P3OUT |= P3_SHIFTOE;
    for (j=0;j<10000;j++)
        _NOP();
    P2OUT |= P2_SHIFTD;
    for (j=0;j<10000;j++)
        _NOP();
}

```

```

P2OUT |= P2_SHIFTCLK;
for (j=0;j<10000;j++)
    _NOP();
if(((P3IN & P3_SENSOR)==0)&&(pill[8]==0))
{
    pillstaken++;
    pill[6]=day;
    pill[7]=hour;
    pill[8]=min+0x80;
}
else if((pillstaken==9)&&((P3IN & P3_SENSOR)>0))
    refill++;
P2OUT &= ~P2_SHIFTCLK;
for (j=0;j<10000;j++)
    _NOP();
P2OUT &= ~P2_SHIFTD;
for (j=0;j<10000;j++)
    _NOP();
for (i=1;i<9;i++)
{
    P2OUT |= P2_SHIFTCLK;
    indoffset=i*3;
    for (j=0;j<10000;j++)
        _NOP();
    if(((P3IN & P3_SENSOR)==0)&&(pill[8+indoffset]==0))
    {
        pillstaken++;
        pill[6+indoffset]=day;
        pill[7+indoffset]=hour;
        pill[8+indoffset]=min+0x80;
    }
    else if((pillstaken==9)&&((P3IN & P3_SENSOR)>0))
        refill++;
    P2OUT &= ~P2_SHIFTCLK;
    for (j=0;j<10000;j++)
        _NOP();
}
for (i=0;i<8;i++)
{
    P2OUT |= P2_SHIFTCLK;
    for (j=0;j<10000;j++)
        _NOP();
    P2OUT &= ~P2_SHIFTCLK;
    for (j=0;j<10000;j++)
        _NOP();
}
P3OUT &= ~P3_SHIFTOE;
if(refill==9)
{
    day=0;
    hour=0;
    min=0;
    for(i=6;i<33;i++)
        pill[i]=0;
    pill[3]=2;
}
}

#pragma vector=PORT2_VECTOR
__interrupt void Port_2(void)
{
    ShiftReg();
    clockadjust();           //Match 6MHz
    //Clear switch interrupt flag
    P2IFG &= ~P2_SWITCH;

    //Clear automatic interrupt flag
    // P2IFG &= ~0x20;

    _BIS_SR(LPM1_bits + GIE);    //Enable MCLK and interrupts

```



```

}

#pragma vector=TIMER0_A0_VECTOR
__interrupt void Timer0_A0(void)
{
    //Switch changes data rate for preamble pulses and data
    switch(premcount)
    {
        case 40: TA0CCR0=252;        //42us pulse
                  break;
        case 41: TA0CCR0=324;        //54us pulse
                  break;
        case 42: TA0CCR0=108;        //18us pulses for data rate
                  break;
    }
    //Checks if its the first half (even or odd premcount) of a data bit for manchester
    if((premcount & 1)==0)
    {
        if((pill[pillind] & (char)bitmask)==0)    //if bit is a 0, typecasted as char since initialized as unsigned int,
                                                    //and only 8 bits needed
            P1OUT &= ~P1_RFDATA;                //make it a rising edge in freq or fm (falling edge in logic)
        else
            P1OUT |= P1_RFDATA;                    //else make it a falling edge
    }
    else
    {
        P1OUT ^= P1_RFDATA;                        //if its the second half, toggle the bit state
        bitmask<<=1;                               //Shift the bitmask
    }
    if(bitmask==0x0200) //check bitmask to see if 9 bits have been passed
    {
        //the 9th will always be a zero due to the typecasting to char leading to bitmask=0x00
        pillind++; //increment the pillind for the pill array since current byte already transmitted
        bitmask=1; //reset the bit mask
        if(pillind==pill[3]+4) //check if last data byte transfered
        {
            TA0CCTL0=0x00;
            for (j=0;j<100;j++)
                _NOP();
            P1OUT &= ~P1_RFCE;                //Disable ADF7020
            if(refill==9)
            {
                pill[3]=29;
                refill=0;
            }
            _BIS_SR(LPM3_bits + GIE);        //Disable MCLK
        }
    }
    premcount++; //Increment premcount
} // end ISR

//Timer1 A0 interrupt service routine
#pragma vector=TIMER1_A0_VECTOR
__interrupt void Timer1_A0 (void)
{
    if (min==59)
    {
        min=0;
        if (hour==23)
        {
            hour=0;
            day++;
        }
        else
            hour++;
    }
    else
    {
        min++;
    }
}

```

```

// Timer0 A2 interrupt service routine
#pragma vector=TIMER0_A1_VECTOR
__interrupt void Timer0_A2 (void)
{
    switch (__even_in_range(TA0IV, 10))    // Efficient switch-implementation
    {
        case 2: break;                    // TA0CCR1 not used
        case 4: clkcapnew=TA0CCR2;          // TA0CCR2 used for DCO/MCLK Calibration
                clkcapdiff=clkcapnew-clkcapold;    //measured timer during 1/(8*ACLK)s
                clkcapold=clkcapnew;
                if (clktarget>clkcapdiff)          //speed up DCO
                {
                    clkindex=0;
                    DCOCTL++;
                }
                else if(clktarget<clkcapdiff)      //slow down DCO
                {
                    clkindex=0;
                    DCOCTL--;
                }
                else
                {
                    clkindex++;                    //Count how many measurments match the target
                    if(clkindex==3)//clock is stalalized
                    {
                        TA0CCTL2 =0x00;    //Disable TA0 CCR2 Interrupts

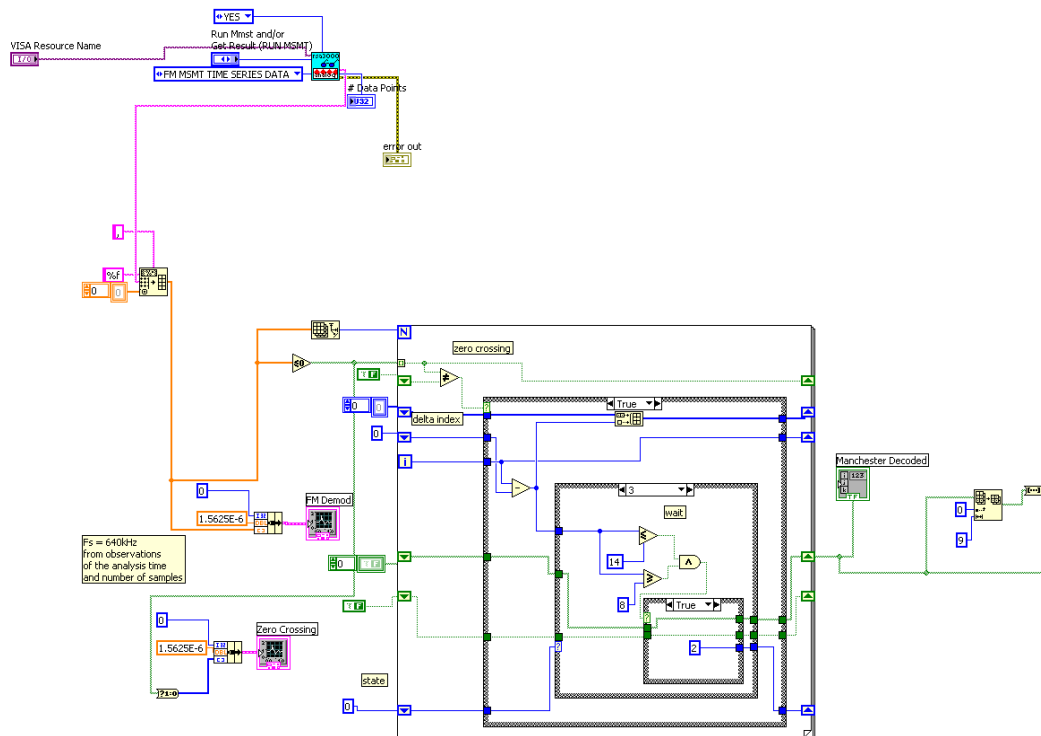
                        P1OUT |= P1_RFCE; //Enable ADF7020
                        while(~P3IN&P3_RFMUX); //Checks that P3.2 (ADF MUXOUT) is high
                                                for regulator check
                        BattCheck();
                        //Configures ADF7020 for TX
                        ADF7020(0x72738000);    //Tuned to 433.92MHz using
                                                RTSA
                        ADF7020(0x000AB011);
                        ADF7020(0xC04A7E12);    //Set for 13dBm
                        ADF7020(0x0DD00376);
                        while(~P3IN&P3_RFMUX); //Checks for ADF Digital Lock
                                                Detect
                        TxClk(); //Enables Tx Timer Interrupt
                    }
                }
                break;
        case 10: break;                    // overflow
    }
}

```

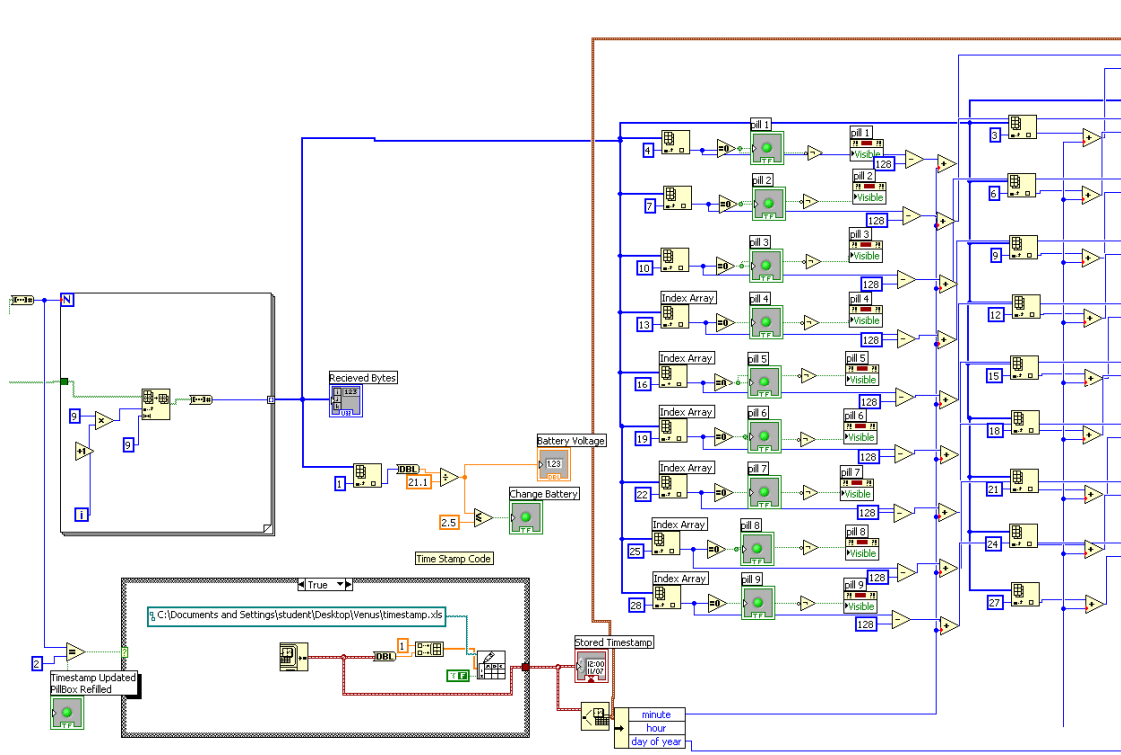
## APPENDIX B

### LABVIEW CODE

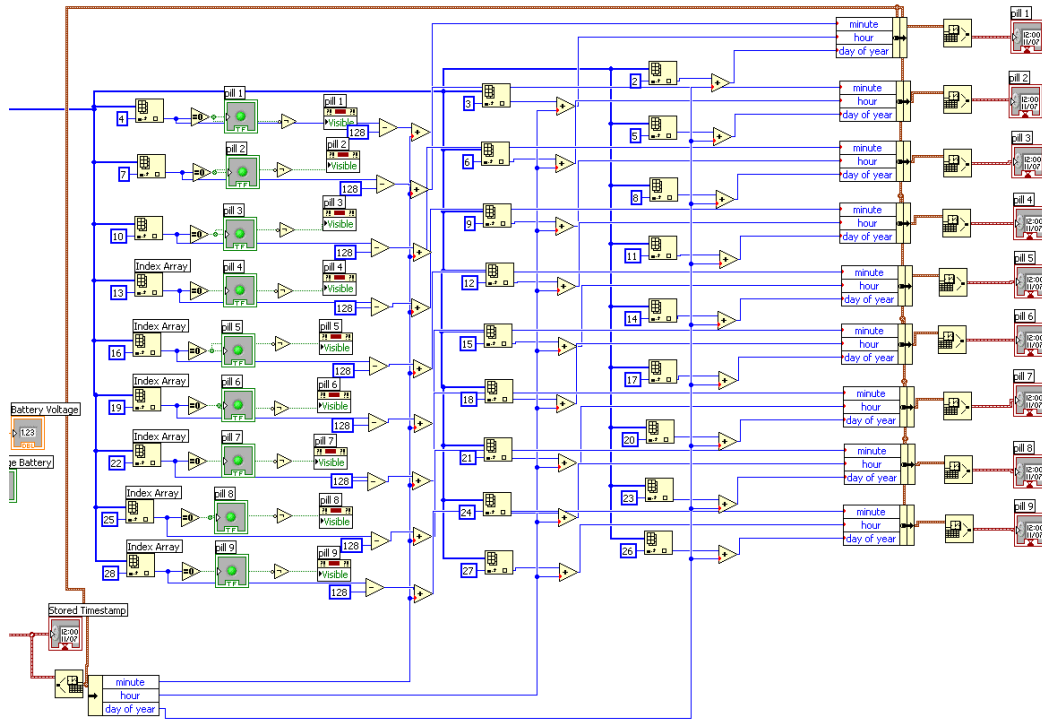
#### B.1 PARSING AND MANCHESTER DECODE



## B.2 REFILL CONDITION AND TIMESTAMP STORAGE



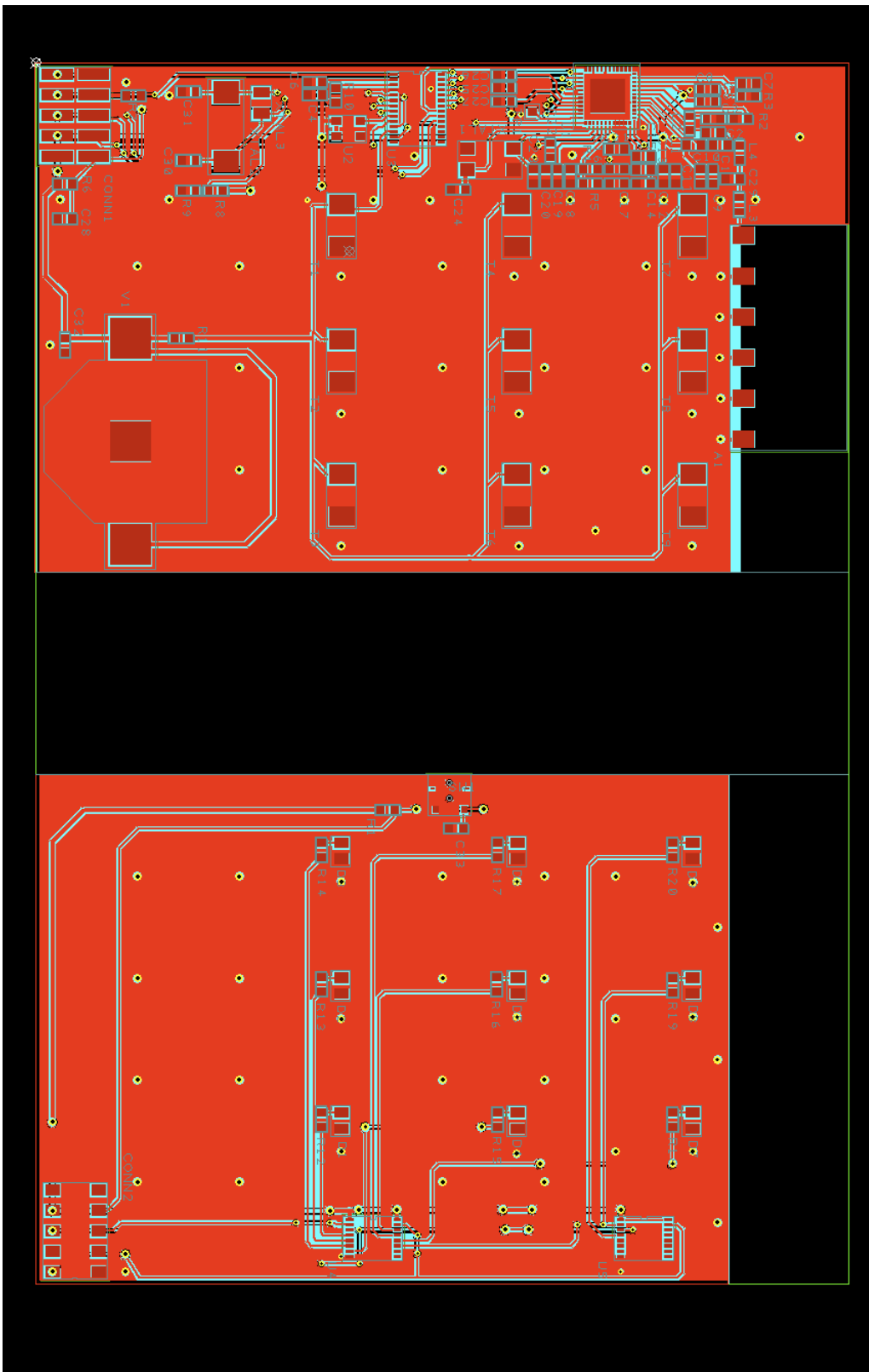
### B.3 TIMESTAMP OFFSET AND DISPLAY



## **APPENDIX C**

### **PCB LAYOUT AND SCHEMATIC**







## BIBLIOGRAPHY

1. J. Connor, N. Rafter, and A. Rodgers, "Do fixed-dose combination pills or unit-of-use packaging improve adherence? A systematic review," *Bull World Health Organ*, vol. 12, pp. 935-939, 2004.
2. S. C. Kalichman, D. Cain, C. Cherry, M. Kalichman, and H. Pope, "Pillboxes and antiretroviral adherence: prevalence of use, perceived benefits, and implications for electronic medication monitoring devices," *AIDS Patient Care STDS*, vol. 19, pp. 833-9, 2005.
3. C. S. Wendel, M. J. Mohler, K. Kroesen, N. M. Ampel, A. L. Gifford, and S. J. Coons, "Barriers to use of electronic adherence monitoring in an HIV clinic," *Ann Pharmacother*, vol. 35, pp. 1010-5, 2001.
4. Med-eMonitor by Informedix,  
<http://www.informedix.com/consumer/ProductDescription.htm>.
5. MD.2 by e-pill,  
[http://www.epill.com/md2.html?source=google&gclid=CP2S\\_NDkzZYCFQVxFQod3ns\\_o2g](http://www.epill.com/md2.html?source=google&gclid=CP2S_NDkzZYCFQVxFQod3ns_o2g).
6. MIT uBox, <http://web.mit.edu/newsoffice/2008/itw-india-tt0206.html>.
7. T. L. Hayes, J. M. Hunt, A. Adami, and J. A. Kaye, "An Electronic Pillbox for Continuous Monitoring of Medication Adherence," *Proceedings of the 27<sup>th</sup> Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 30 August – 3 September, New York, NY, 2006.
8. Medication Event Monitoring System (MEMS) by Aardex  
[http://www.aardexgroup.com/aardex\\_index.php?group=aardex&id=85](http://www.aardexgroup.com/aardex_index.php?group=aardex&id=85).
9. A. Muehlhofer, "Controlling the DCO Frequency of the MSP430x11x," *Texas Instruments Deutschland, Application Report SLAA074*, April 2000.
10. M. Mitchell, "Implementing a Real-Time Clock on the MSP430," *Texas Instruments Mixed Signal Products, Application Report SLAA076A*, January 2001.
11. "MSP430x2xx Family User's Guide", *Texas Instruments Mixed Signal Products, SLAU144E*, 2008.

12. "MSP-FET430 Flash Emulation Tool (FET) (for Use With Code Composer Essentials for MSP430 Version 3) User's Guide," Texas Instruments, SLAU157G, May 2008.
13. J. Bachiochi, "MSP430 News Flash Recognizing the Flexibility of Reprogramming," Circuit Cellar, Issue 133, pp. 68-72, August 2001.
14. "High Performance, ISM Band, FSK/ASK Transceiver IC ADF7020," Analog Devices, Product Data Sheet, Rev. B, 2007.
15. "LG100MN1MPx Surface Mount Type, High Power Output Infrared Emitting Diode," Sharp Corporation, Product Data Sheet, D1-A00701EN, 30 November 2007.
16. "MSP430F21x2 Mixed Signal Microcontroller," Texas Instruments, Product Data Sheet, SLAS568B, May 2008.
17. "SMD Silicon Phototransistor OP520, OP521," Optek Technology Inc., Product Data Sheet, February 2007.
18. "Evaluation Boards for ADF7020 & ADF7020-1 ISM Band Transceiver EVAL-ADF7020-XDBX," Analog Devices, Preliminary Technical Data, Rev. PR D, 16 October, 2007.
19. ADIismLINK Installmode, that is with the hardware unconnected to calculate need to find actual program name
20. A. Harney and G. Retz, "ADF7020 RF Port Impedance Values for Matching Purposes," Analog Devices, Application Note, AN-764, Rev. 0, 2005.
21. K. L. Silvers, J. R. Skorpik, B. J. Burghard, M. A. Hughes, K. E. Dorow, L. J. Kiriara, and F. J. Mauss, "PM J-AIT ISO/IEC 18000-7:2008 Reference Design Model Version 1.0," Pacific Northwest National Laboratory operated by Battelle for the US Department of Energy under contract DE-AC05-76RL01830, 29 February 2008.
22. "Information technology AIDC techniques-Radio frequency identification for item management-Air interface, Part 7: Parameters for active air interface communications at 433MHz (Revision)," Committee identification: ISO/IEC JTC 1/SC31/WG 4, Reference number of working document: ISO/IEC JTC 1/SC 31 N2610, Reference number of document: ISO/IEC FCD 18000-7.2, 20 August, 2008.
23. "Manganese Dioxide Lithium Coin Batteries: Individual Specifications CR2032," Panasonic, Lithium Handbook, February 2002.